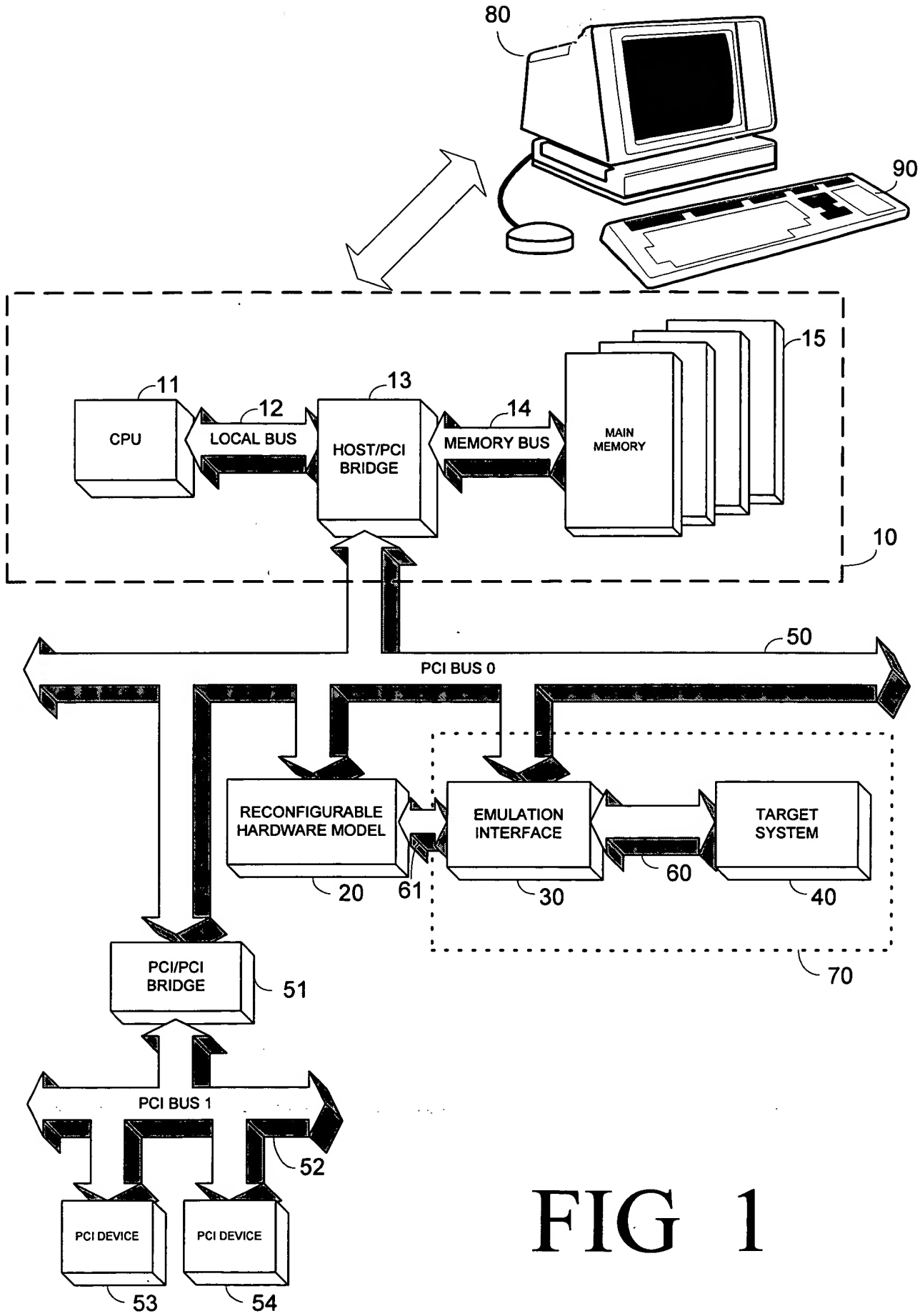


FIG 1

USAGE

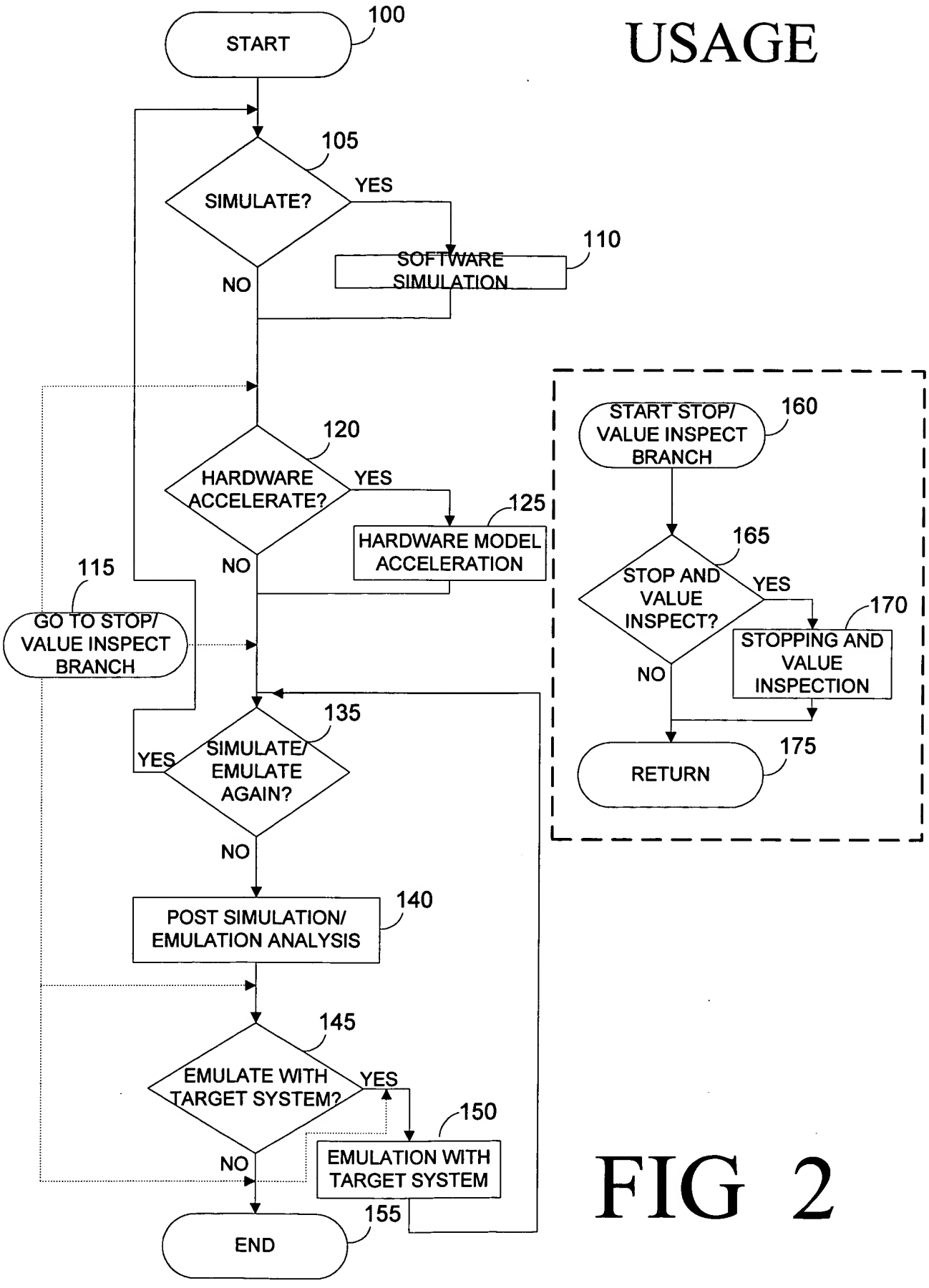


FIG 2

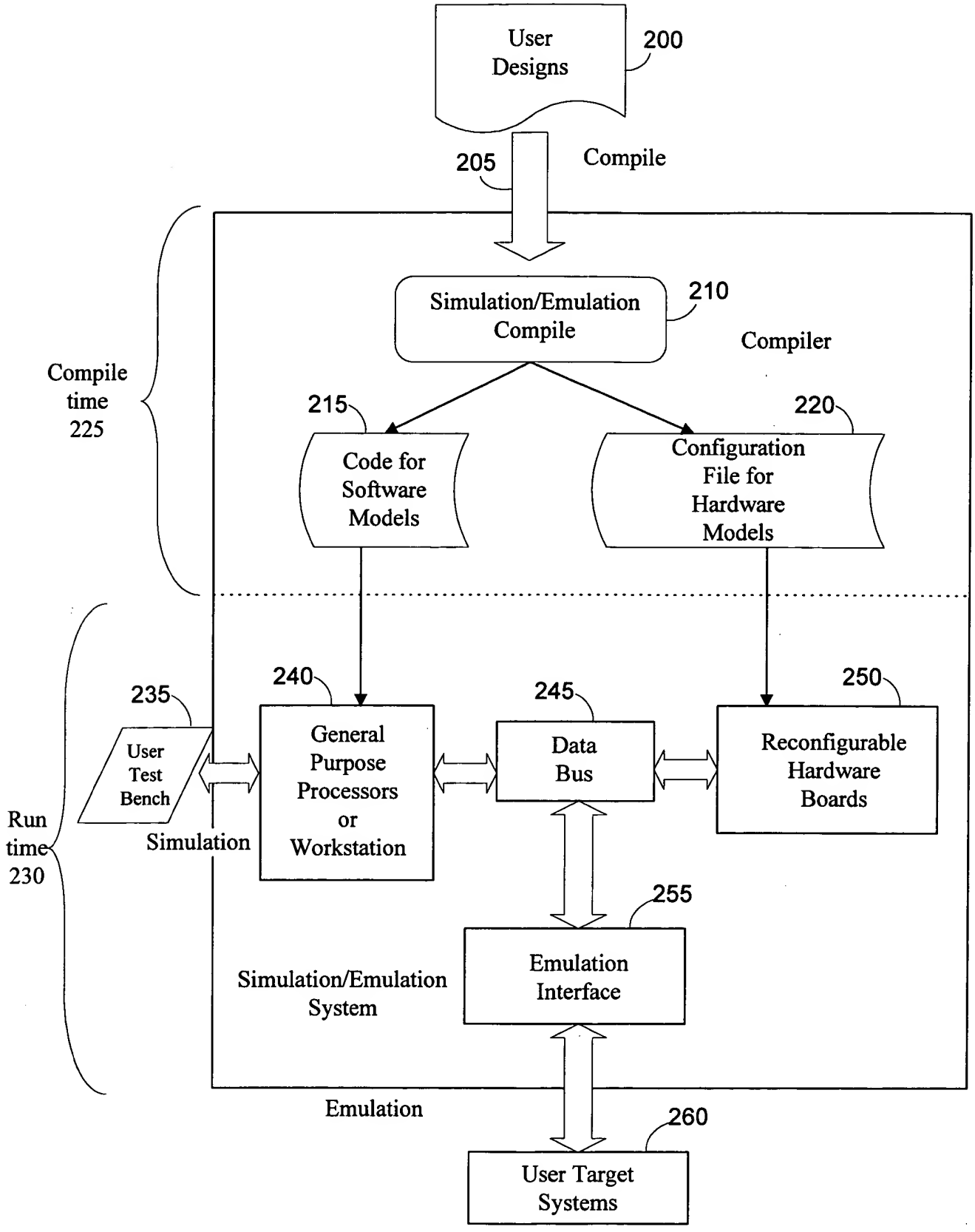

User Designs
200
205
Compile
Simulation/Emulation Compile
210
Compiler
Compile time 225
215
220
Code for Software Models
Configuration File for Hardware Models
235
240
245
250
User Test Bench
General Purpose Processors or Workstation
Data Bus
Reconfigurable Hardware Boards
Run time 230
Simulation
255
Simulation/Emulation System
Emulation Interface
Emulation
260
User Target Systems

FIG 3

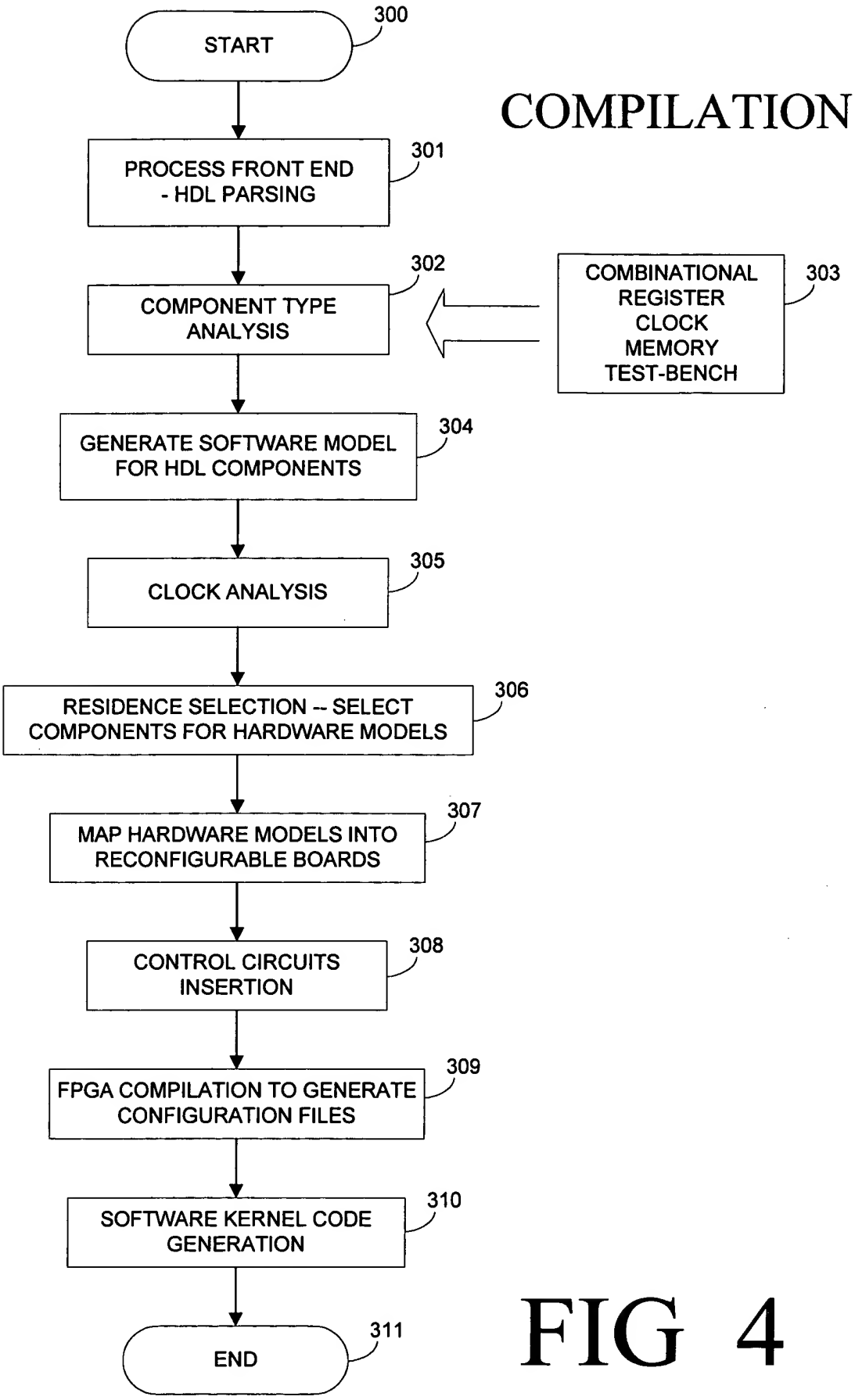
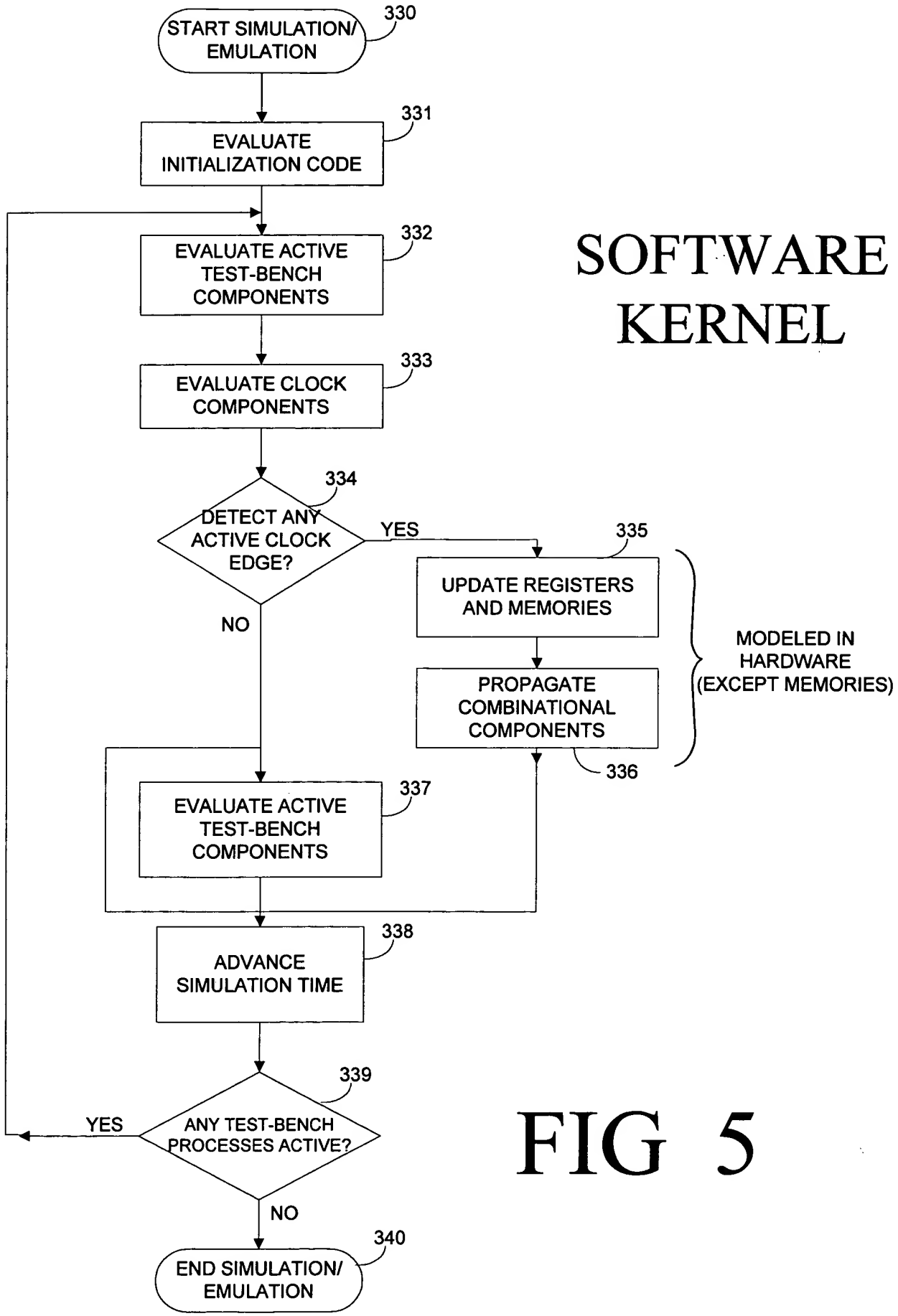

COMPILATION
START
300
PROCESS FRONT END
- HDL PARSING
301
COMPONENT TYPE
ANALYSIS
302
COMBINATIONAL
REGISTER
CLOCK
MEMORY
TEST-BENCH
303
GENERATE SOFTWARE MODEL
FOR HDL COMPONENTS
304
CLOCK ANALYSIS
305
RESIDENCE SELECTION -- SELECT
COMPONENTS FOR HARDWARE MODELS
306
MAP HARDWARE MODELS INTO
RECONFIGURABLE BOARDS
307
CONTROL CIRCUITS
INSERTION
308
FPGA COMPILATION TO GENERATE
CONFIGURATION FILES
309
SOFTWARE KERNEL CODE
GENERATION
310
END
311

FIG 4



SOFTWARE KERNEL

FIG 5

MAPPING HARDWARE MODELS TO RECONFIGURABLE BOARDS

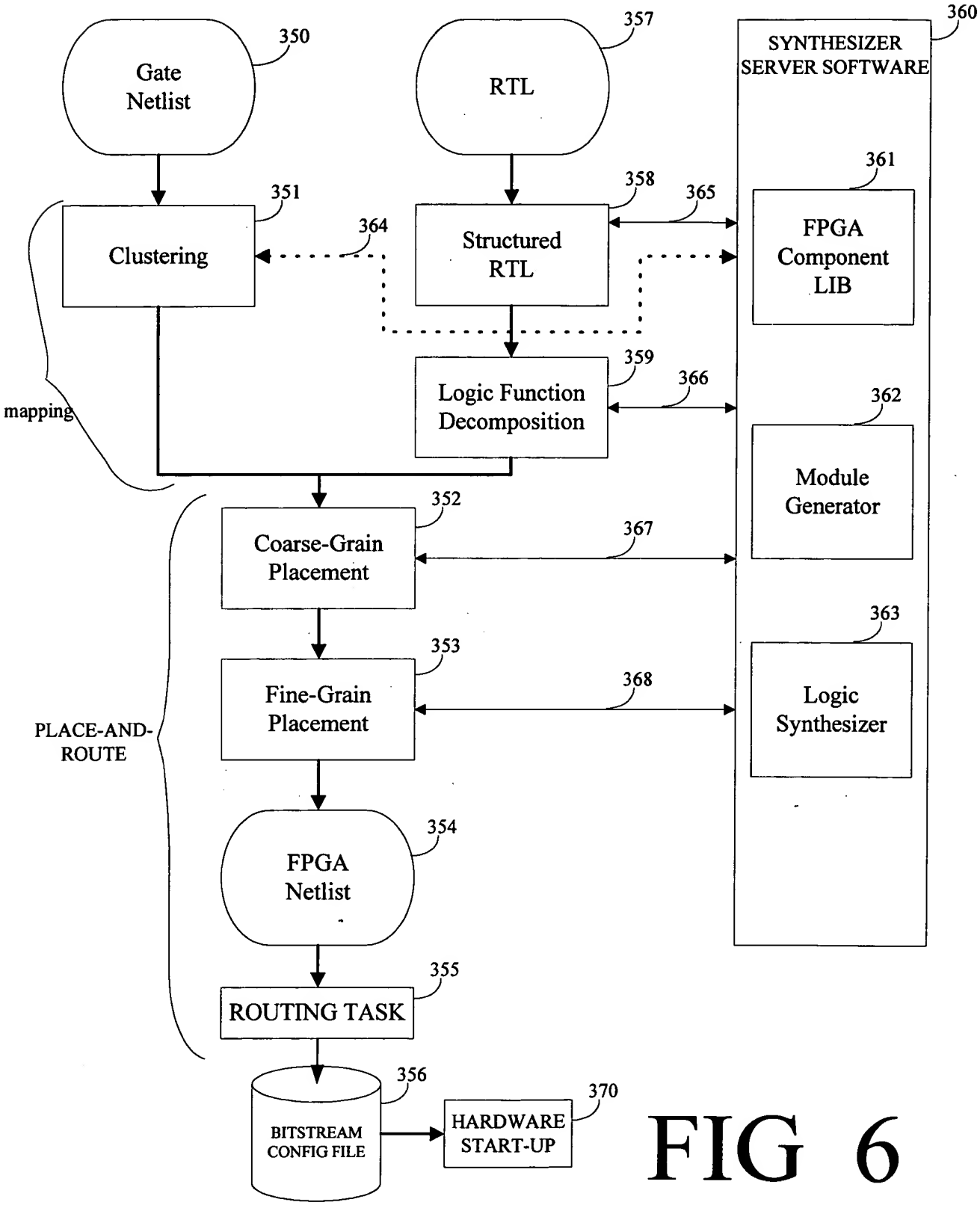


FIG 6

	F11	F12	F13	F14	F21	F22	F23	F24	F31	F32	F33	F34	F41	F42	F43	F44
F11	1	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0
F12	1	1	1	1	0	1	0	0	0	1	0	0	0	1	0	0
F13	1	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0
F14	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1
F21	0	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0
F22	1	1	0	0	1	1	1	1	0	1	0	0	0	1	0	0
F23	0	0	1	0	1	1	1	1	0	0	1	0	0	0	1	0
F24	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1
F31	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0
F32	1	1	0	0	0	1	0	0	1	1	1	1	0	1	0	0
F33	0	0	1	0	0	0	1	0	1	1	1	1	0	0	1	0
F34	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
F41	0	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1
F42	1	1	0	0	0	1	0	0	0	1	0	0	1	1	1	1
F43	0	0	1	0	0	0	1	0	0	0	1	0	1	1	1	1
F44	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	1

FIG. 7

FPGA INTERCONNECTION

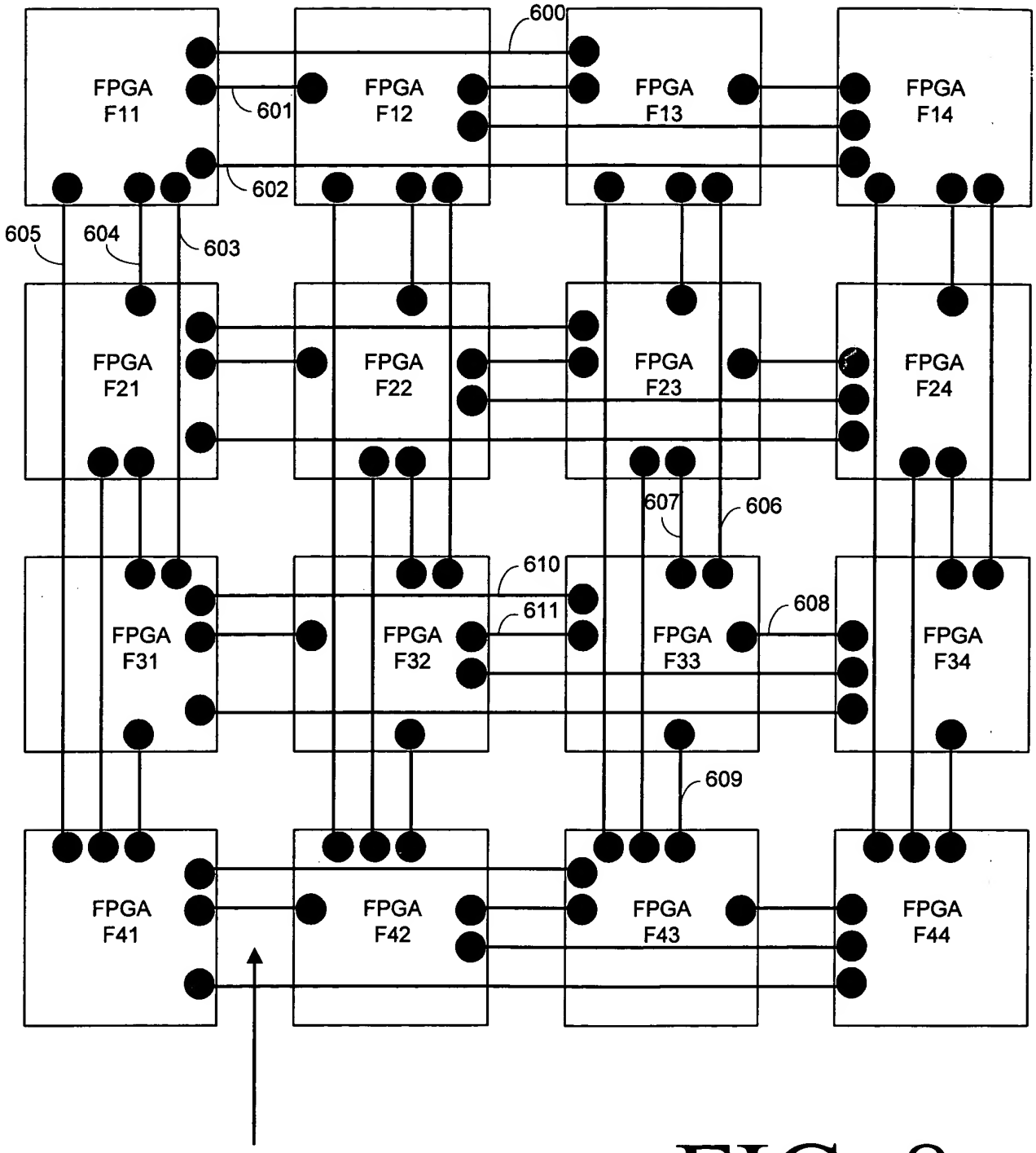
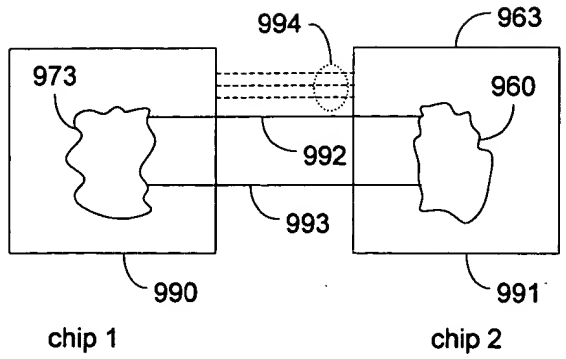
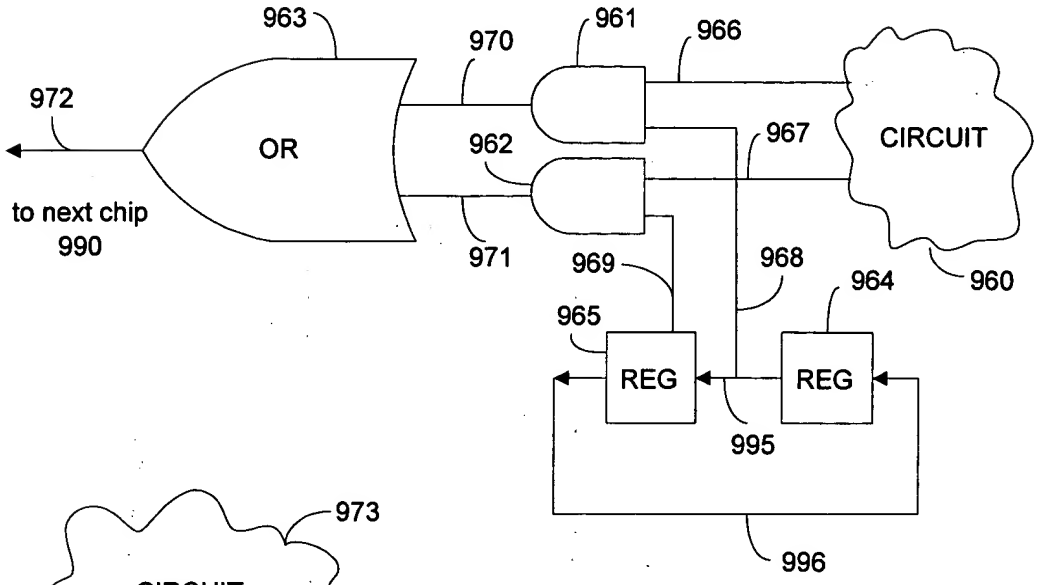


FIG 8

(A)



(B)



(C)

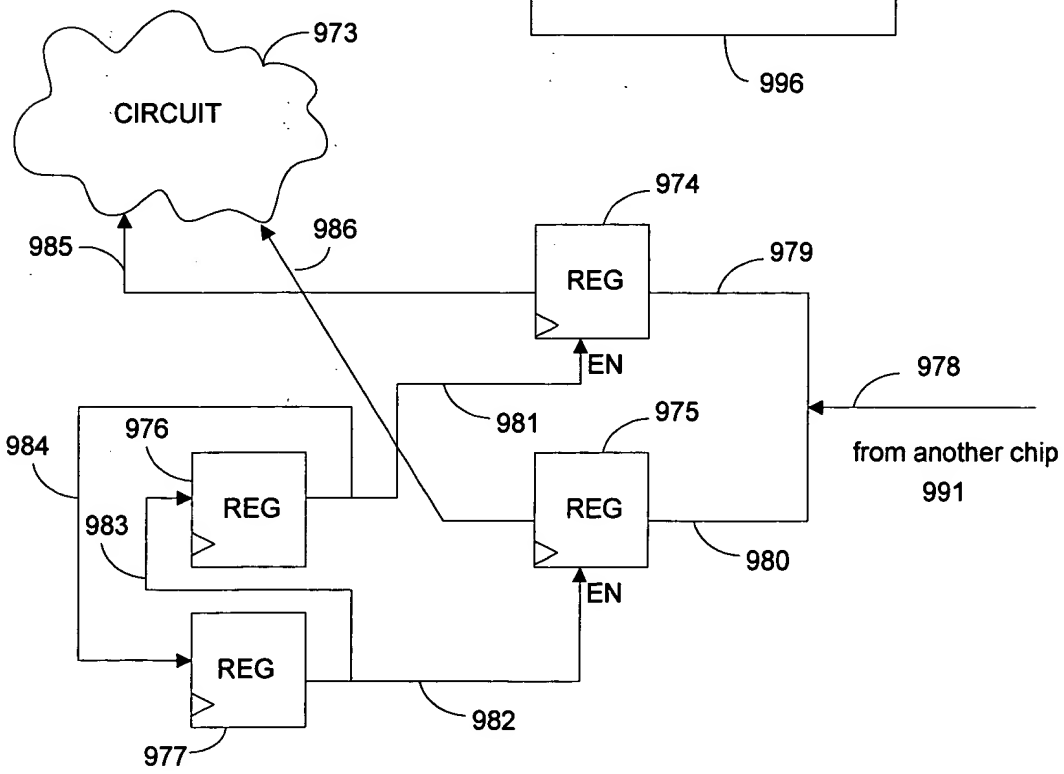


FIG 9

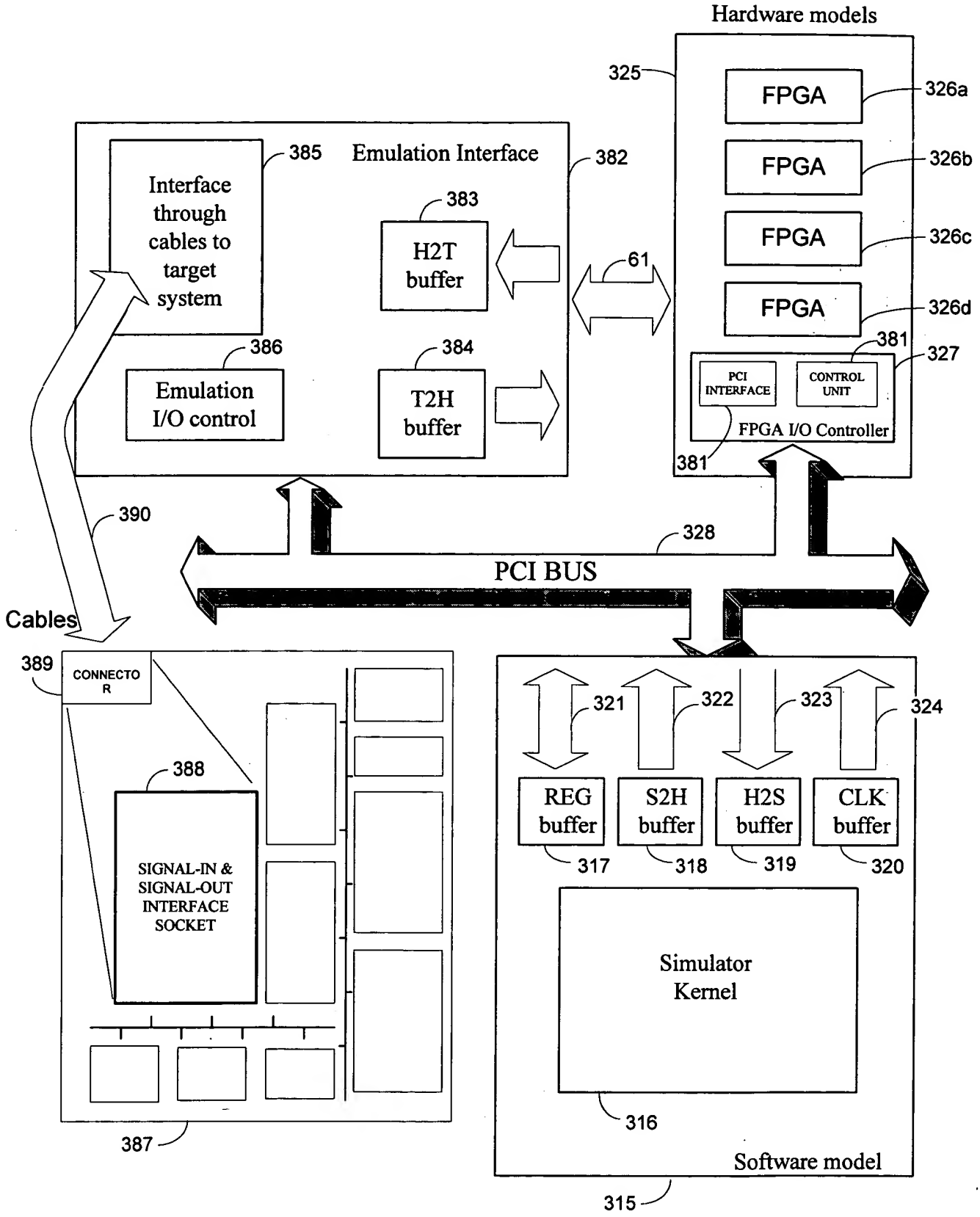

Hardware models
325
FPGA
326a
FPGA
326b
FPGA
326c
FPGA
326d
381
327
PCI INTERFACE
CONTROL UNIT
FPGA I/O Controller
381
385
Emulation Interface
382
Interface through cables to target system
383
H2T buffer
61
386
Emulation I/O control
384
T2H buffer
390
328
PCI BUS
Cables
389
CONNECTOR
388
SIGNAL-IN & SIGNAL-OUT INTERFACE SOCKET
387
321
322
323
324
REG buffer
S2H buffer
H2S buffer
CLK buffer
317
318
319
320
Simulator Kernel
316
Software model
315

FIG 10

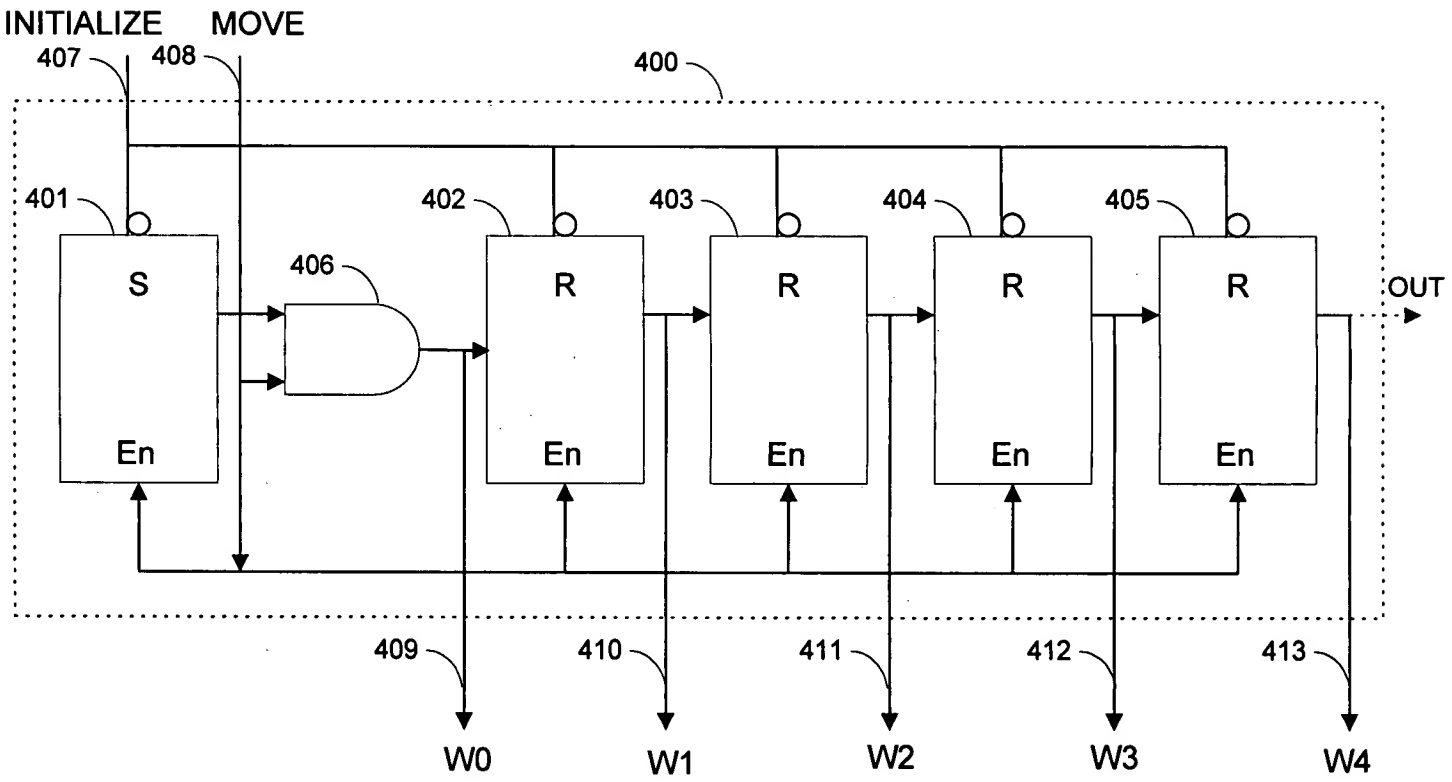

INITIALIZE
MOVE
407
408
400
401
402
403
404
405
S
R
R
R
R
406
OUT
En
En
En
En
En
409
410
411
412
413
W0
W1
W2
W3
W4

FIG. 11

ADDRESS POINTER
INITIALIZATION

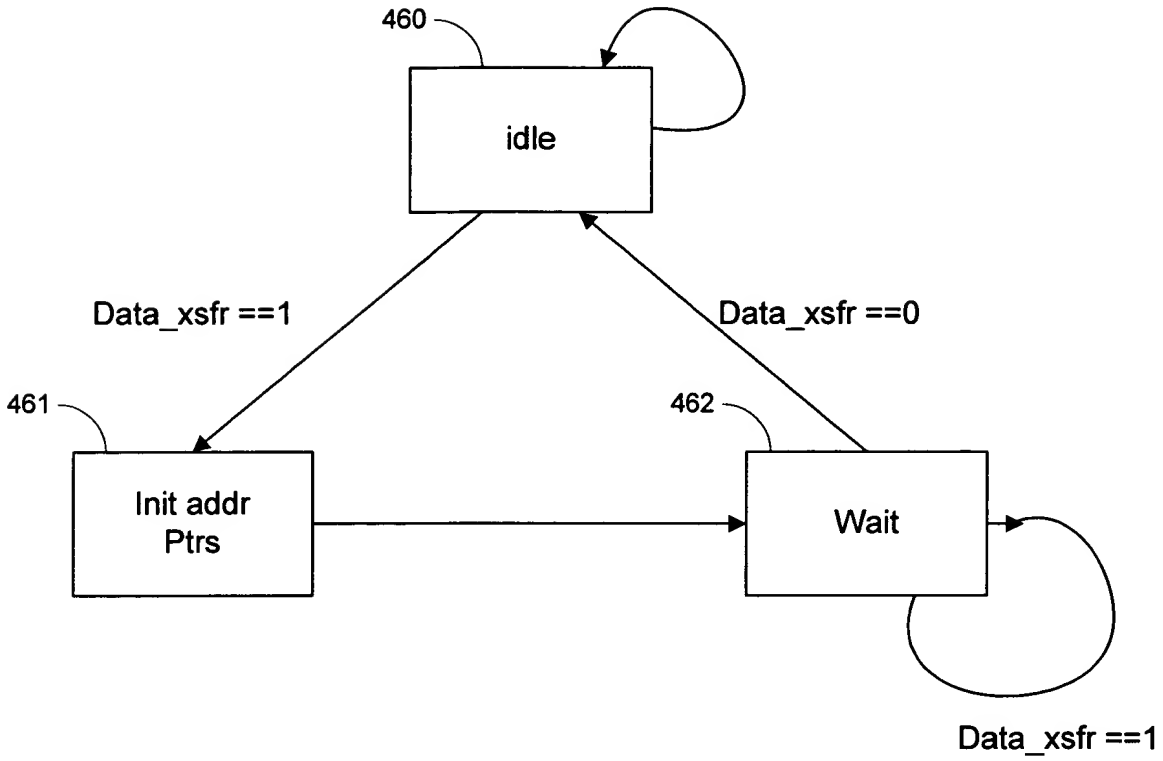


FIG. 12

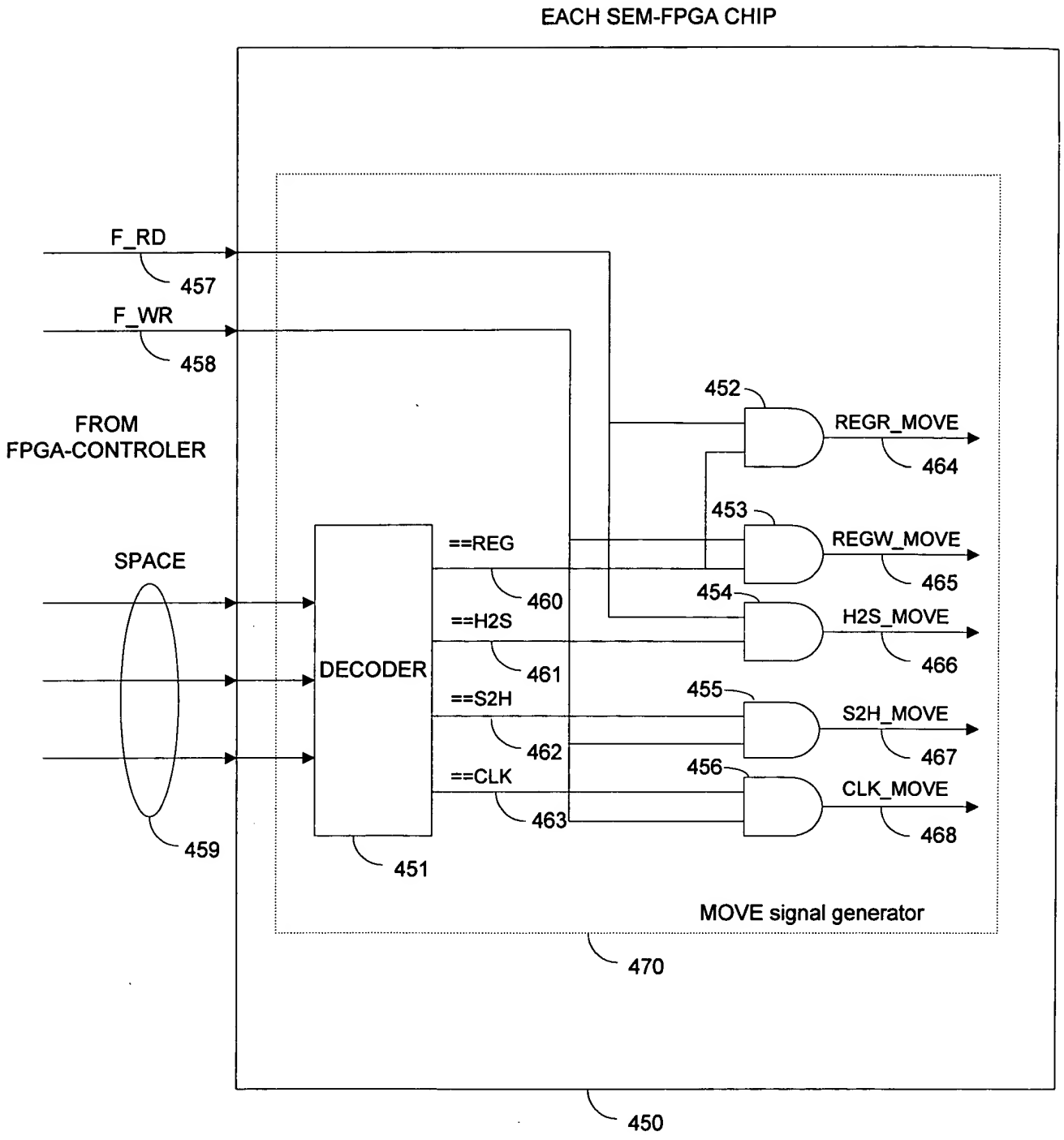

EACH SEM-FPGA CHIP
F_RD
457
F_WR
458
FROM
FPGA-CONTROLER
SPACE
459
DECODER
451
==REG
460
==H2S
461
==S2H
462
==CLK
463
452
REGR_MOVE
464
453
REGW_MOVE
465
454
H2S_MOVE
466
455
S2H_MOVE
467
456
CLK_MOVE
468
MOVE signal generator
470
450

FIG 13

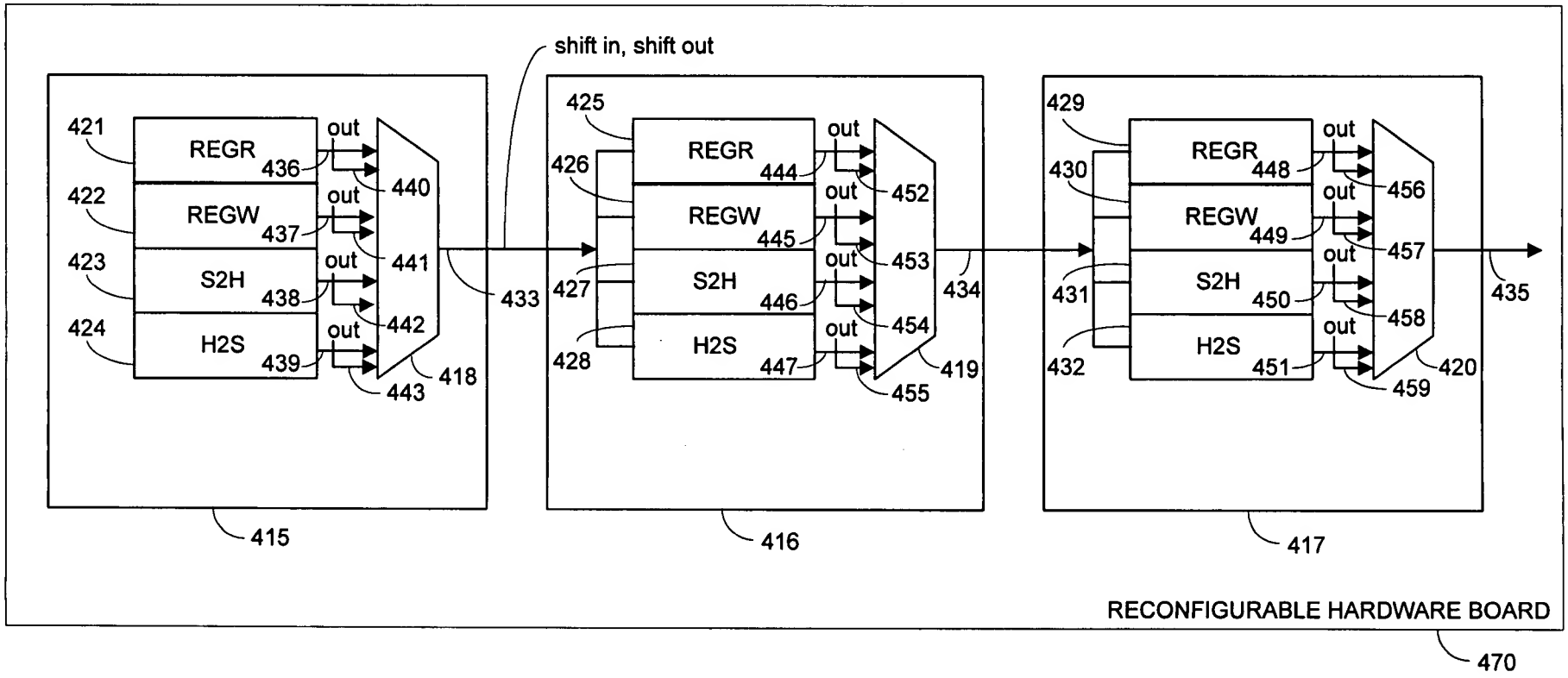

shift in, shift out
421
REGR
436
out
422
REGW
437
out
423
S2H
438
out
424
H2S
439
out
440
441
442
443
418
433
425
426
REGR
444
out
REGW
445
out
427
S2H
446
out
428
H2S
447
out
452
453
454
455
419
434
429
430
REGR
448
out
REGW
449
out
431
S2H
450
out
432
H2S
451
out
456
457
458
459
420
435
415
416
417
RECONFIGURABLE HARDWARE BOARD
470

FIG. 14

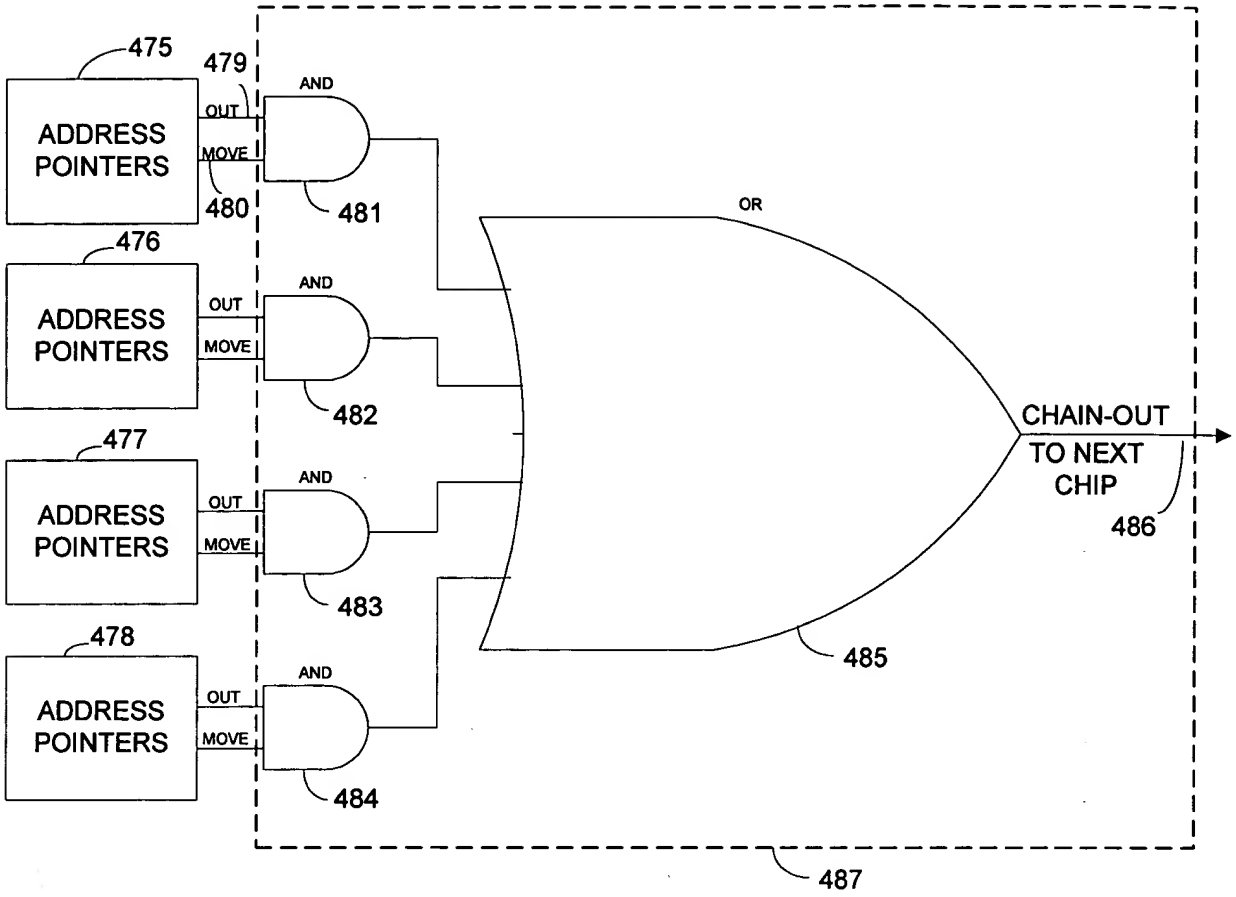

475
479
ADDRESS POINTERS
OUT
MOVE
480
AND
481
OR
476
ADDRESS POINTERS
OUT
MOVE
AND
482
CHAIN-OUT
TO NEXT CHIP
486
477
ADDRESS POINTERS
OUT
MOVE
AND
483
485
478
ADDRESS POINTERS
OUT
MOVE
AND
484
487

FIG. 15

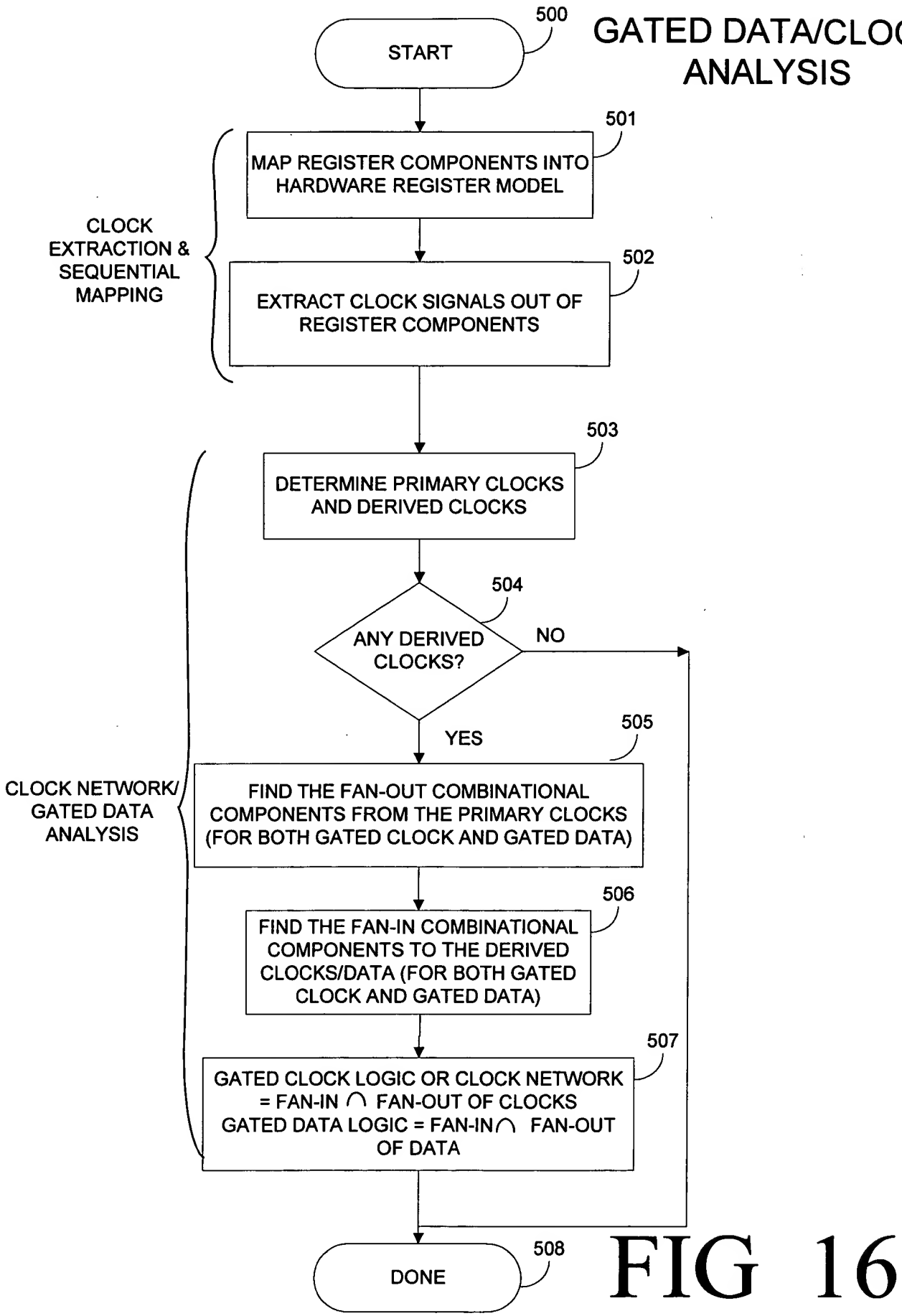

GATED DATA/CLOCK ANALYSIS
500
START
501
MAP REGISTER COMPONENTS INTO HARDWARE REGISTER MODEL
CLOCK EXTRACTION & SEQUENTIAL MAPPING
502
EXTRACT CLOCK SIGNALS OUT OF REGISTER COMPONENTS
503
DETERMINE PRIMARY CLOCKS AND DERIVED CLOCKS
504
ANY DERIVED CLOCKS?
NO
YES
505
CLOCK NETWORK/ GATED DATA ANALYSIS
FIND THE FAN-OUT COMBINATIONAL COMPONENTS FROM THE PRIMARY CLOCKS (FOR BOTH GATED CLOCK AND GATED DATA)
506
FIND THE FAN-IN COMBINATIONAL COMPONENTS TO THE DERIVED CLOCKS/DATA (FOR BOTH GATED CLOCK AND GATED DATA)
507
GATED CLOCK LOGIC OR CLOCK NETWORK = FAN-IN ∩ FAN-OUT OF CLOCKS
GATED DATA LOGIC = FAN-IN∩ FAN-OUT OF DATA
508
DONE

FIG 16

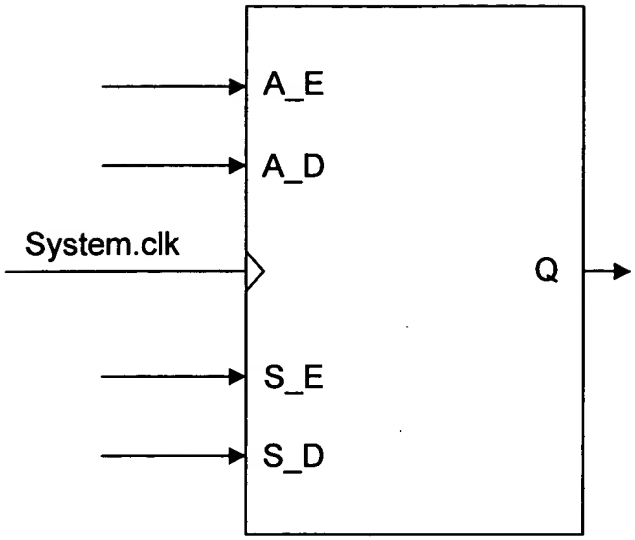
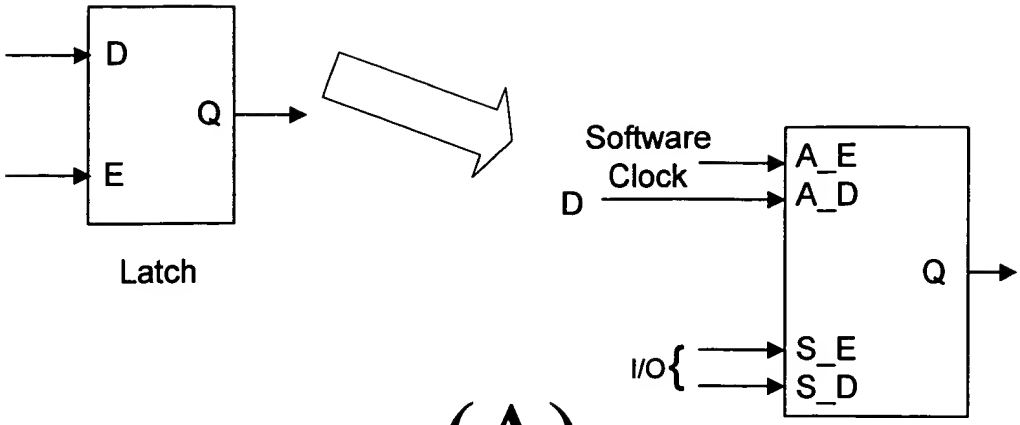
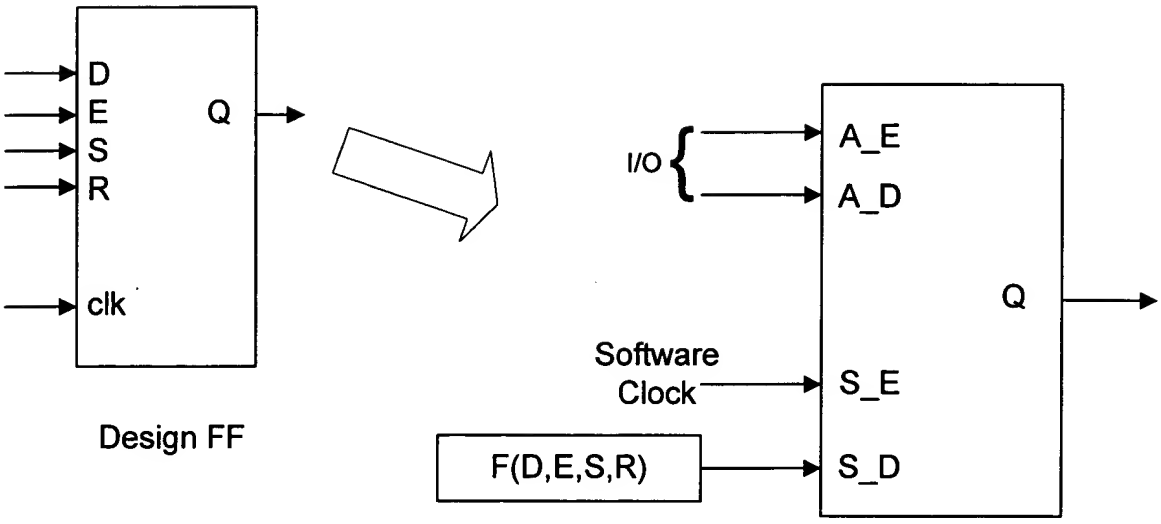


FIG. 17


D
Q
E
Latch
Software
Clock
D
A_E
A_D
Q
I/O
S_E
S_D

(A)


D
E
S
R
Q
clk
Design FF
I/O
A_E
A_D
Q
Software
Clock
S_E
F(D,E,S,R)
S_D

(B)

FIG 18

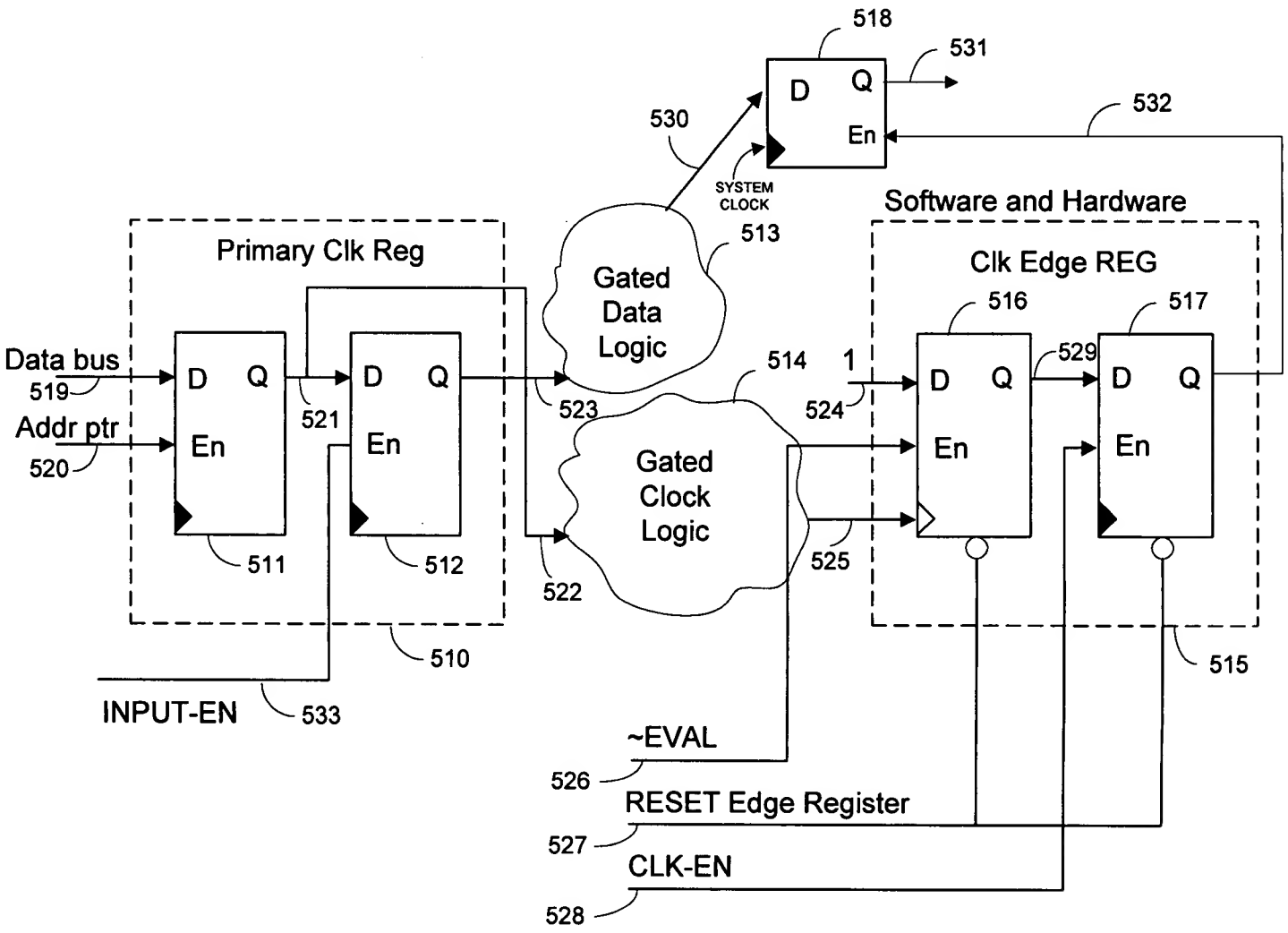

518
531
532
530
D
Q
En
SYSTEM CLOCK
Software and Hardware
Primary Clk Reg
513
Gated Data Logic
Clk Edge REG
516
517
Data bus
519
Addr ptr
520
D
Q
En
521
511
512
523
514
1
524
529
522
Gated Clock Logic
525
510
515
INPUT-EN
533
~EVAL
526
RESET Edge Register
527
CLK-EN
528

FIG 19

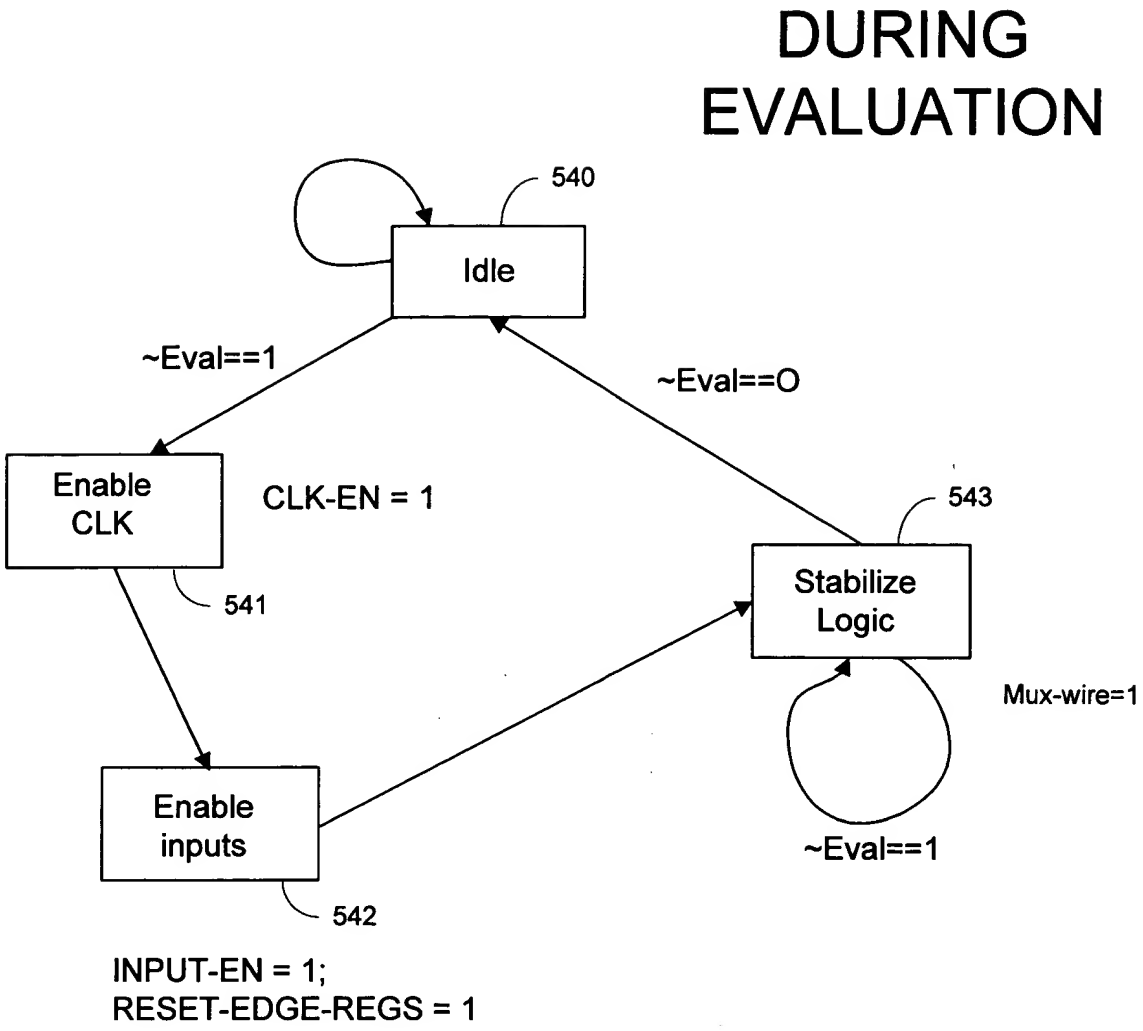

DURING EVALUATION
540
Idle
~Eval==1
~Eval==O
Enable CLK
CLK-EN = 1
543
541
Stabilize Logic
Mux-wire=1
Enable inputs
~Eval==1
542
INPUT-EN = 1;
RESET-EDGE-REGS = 1

FIG. 20

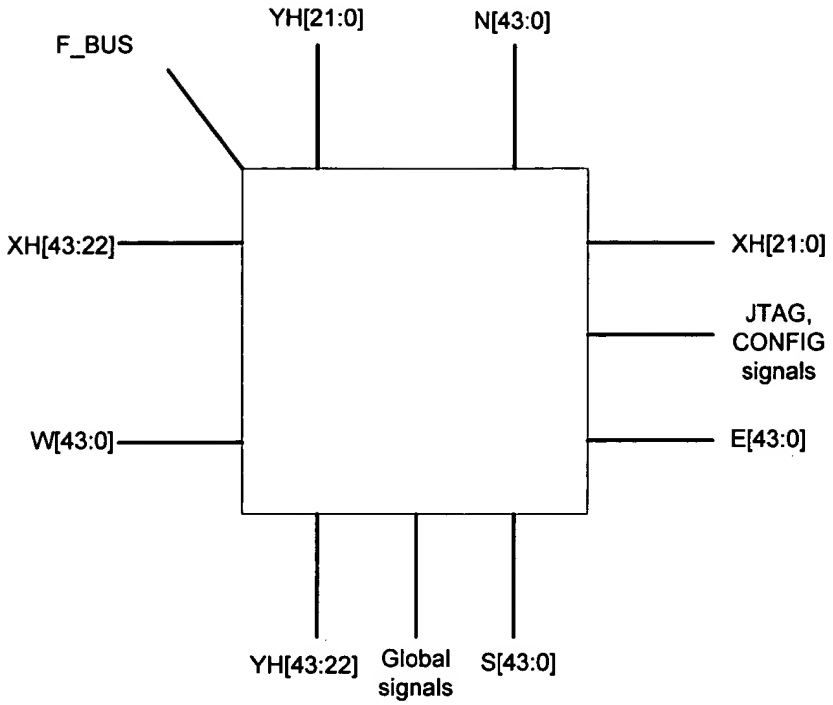


FIG. 21

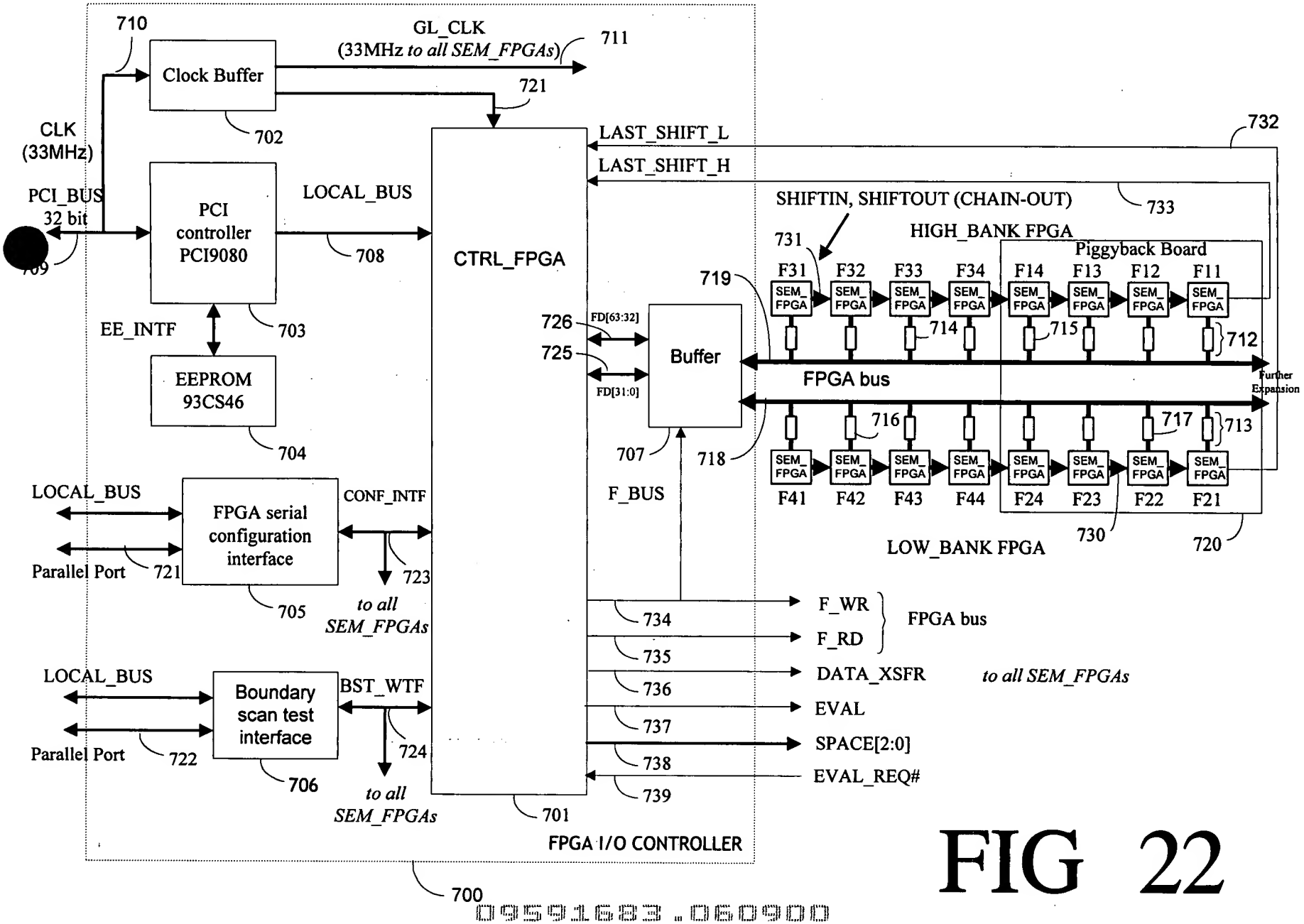


FIG 22

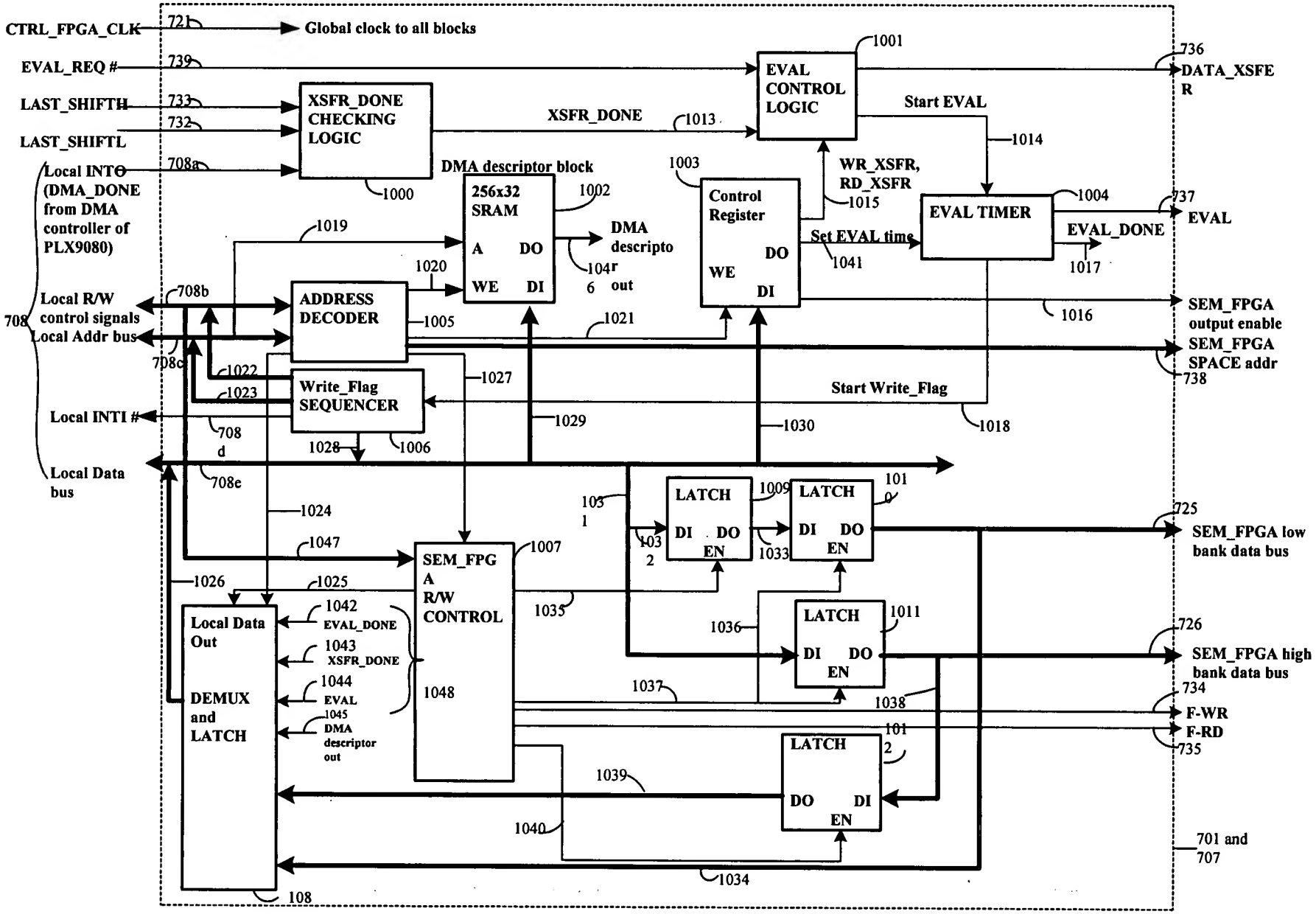

CTRL_FPGA_CLK
721
Global clock to all blocks
EVAL_REQ #
739
LAST_SHIFTH
733
LAST_SHIFTL
732
Local INTO (DMA_DONE from DMA controller of PLX9080)
708a
XSFR_DONE CHECKING LOGIC
1000
XSFR_DONE
1013
EVAL CONTROL LOGIC
1001
736
DATA_XSFER
Start EVAL
1014
WR_XSFR, RD_XSFR
1015
1003
Control Register
DO
WE
DI
Set EVAL time
1041
EVAL TIMER
1004
737
EVAL
EVAL_DONE
1017
DMA descriptor block
256x32 SRAM
A
DO
WE
DI
1002
DMA descriptor out
1046
1019
1020
708
Local R/W control signals
Local Addr bus
708b
ADDRESS DECODER
1005
1021
1016
SEM_FPGA output enable
SEM_FPGA SPACE addr
738
708c
1022
1023
Write_Flag SEQUENCER
1027
Start Write_Flag
1018
Local INTI #
708 d
1028
1006
1029
1030
Local Data bus
708e
1024
1047
1026
1025
SEM_FPGA R/W CONTROL
1007
1048
1035
LATCH
1009
1031
1032
1033
1010
725
SEM_FPGA low bank data bus
LATCH
1011
1036
726
SEM_FPGA high bank data bus
1037
1038
734
F-WR
F-RD
735
LATCH
1012
1039
1040
Local Data Out
DEMUX and LATCH
1042 EVAL_DONE
1043 XSFR_DONE
1044 EVAL
1045 DMA descriptor out
108
1034
701 and 707

FIG. 23

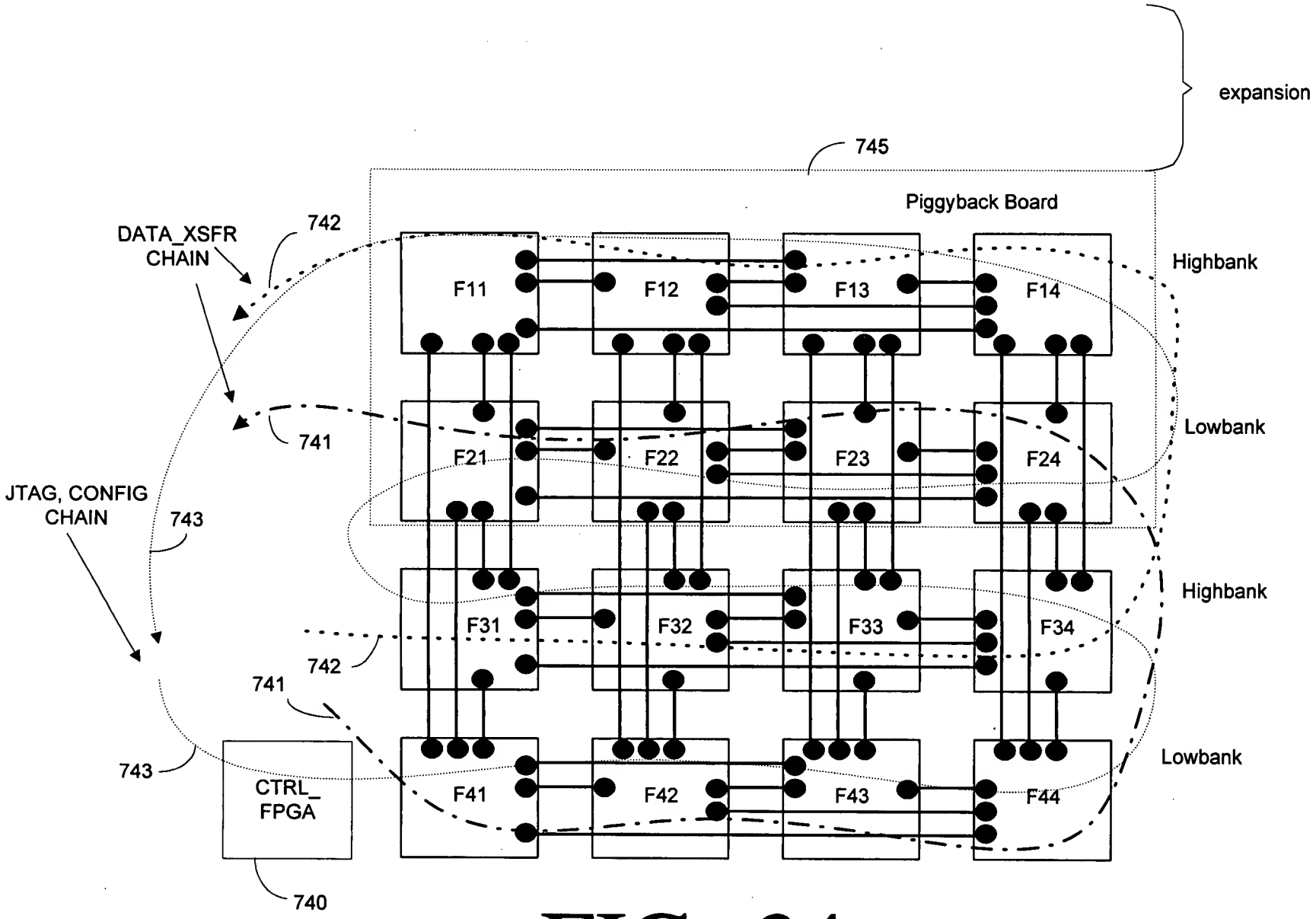
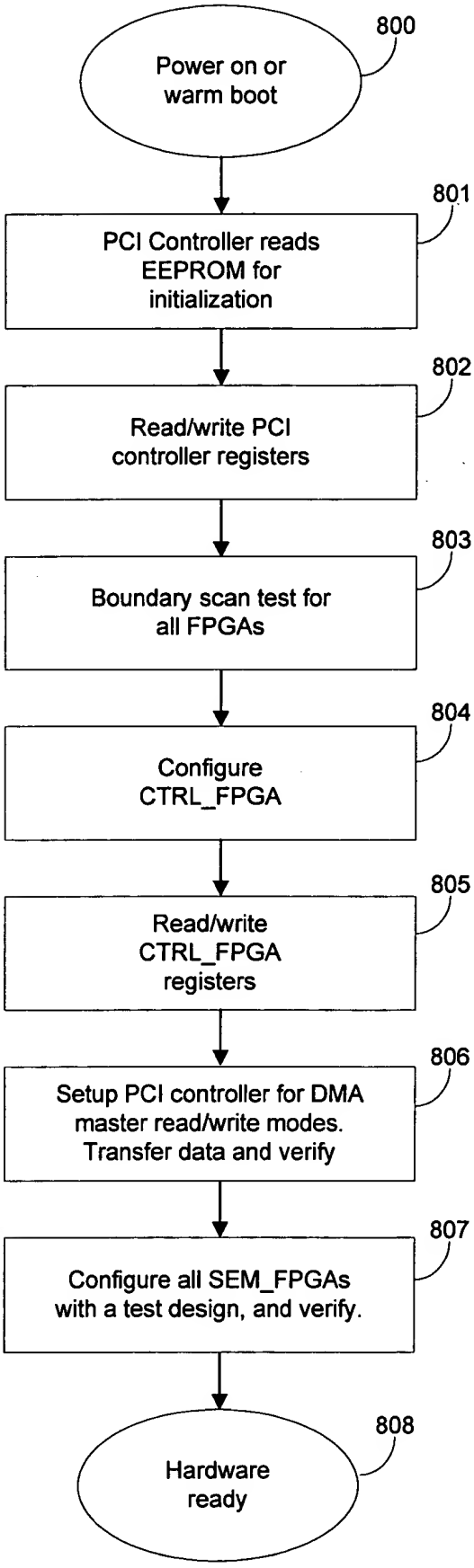


FIG. 24



HARDWARE START-UP

FIG 25

```
module register (clock, reset, d, q);
input clock, d, reset;
output q;
reg q;

always@(posedge clock or negedge reset)
    if(!reset)
        q = 0;
    else
        q = d;

endmodule

module example;
    wire d1, d2, d3;
    wire q1, q2, q3;

    reg sigin;
    wire sigout;
    reg clk, reset;

    register reg1 (clk, reset, d1, q1);
    register reg2 (clk, reset, d2, q2);
    register reg3 (clk, reset, d3, q3);

    assign d1 = sigin ^ q3;
    assign d2 = q1 ^ q3;
    assign d3 = q2 ^ q3;
    assign sigout = q3;

    // a clock generator
    always
    begin
        clk = 0;
        #5;
        clk = 1;
        #5;
    end

    // a signal generator
    always
    begin
        #10;
        sigin = $random;
    end

    // initialization
    initial
    begin
        reset = 0;
        sigin = 0;
        #1;
        reset =1;
        #5;
        $monitor($time, " %b, %b", sigin, sigout);
        #1000 $finish;
    end
    end module
```

FIG. 26

CIRCUIT DIAGRAM

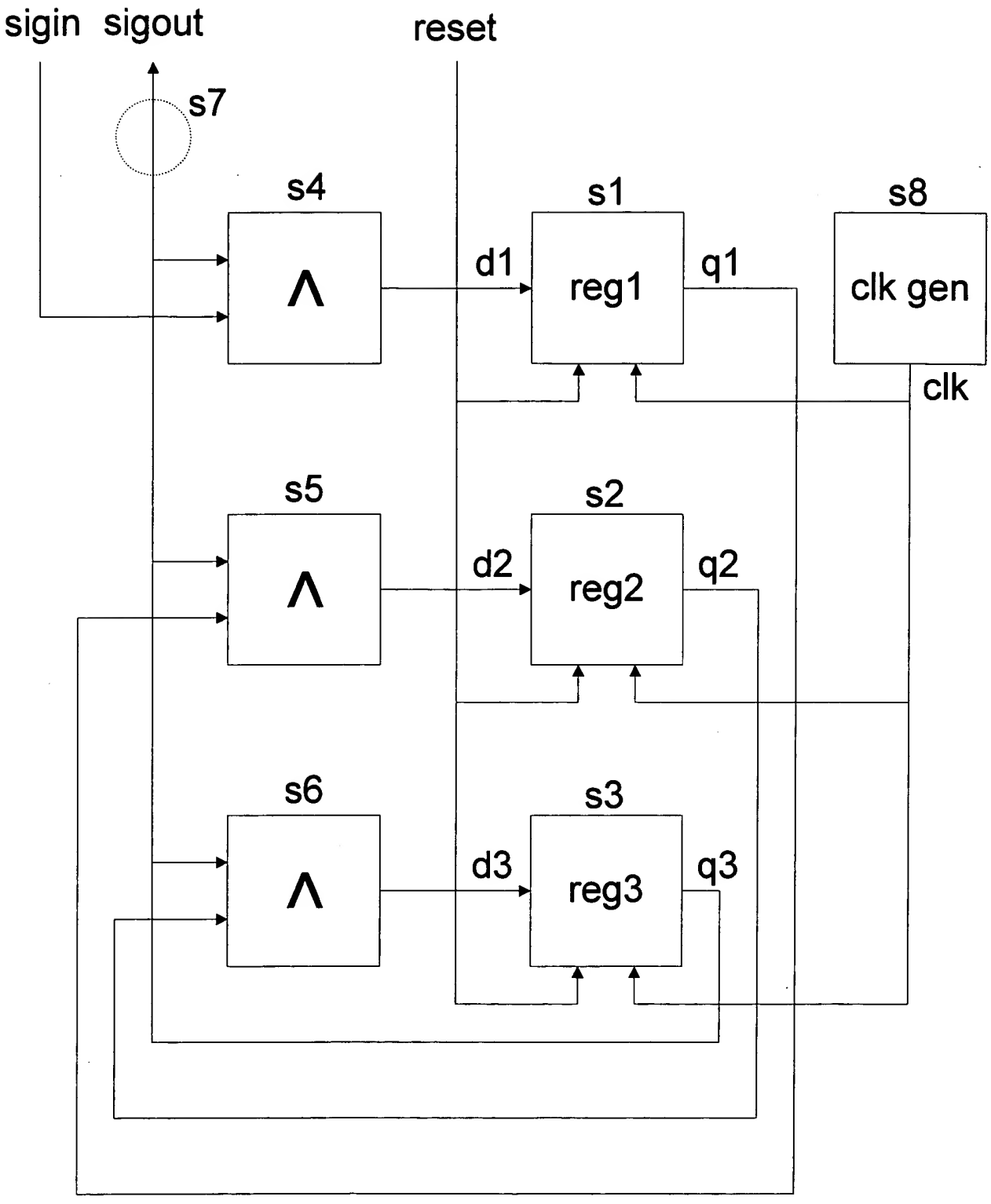


FIG 27

```
module register (clock, reset, d, q);
input clock, d, reset;
output q;
reg q;

always@(postedge clock or negedge reset)
   if(~reset)
      q = 0
   else
      q = d;

endmodule
```

Register Definition

900

```
module example;
   wire d1, d2, d3;
   ware q1, q2, q3;
```

wire interconnection info

907

```
   reg sigin;
   wire sigout;
   reg clk, reset;
```

Test-bench input -- 908

Test-bench output -- 909

```
S1  register reg 1 (clk, reset, d1, q1);
S2  register reg 2 (clk, reset, d2, q2);
S3  register reg 3 (clk, reset, d3, q3);
```

Register component

901

```
S4  assign d1 = sigin ^ q3;
S5  assign d2 = q1 ^ 3;
S6  assign d3 = q2 ^ q3;
S7  assign signout = q3;
```

Combinational component

902

```
S8    // a clock generator
      always
      begin
         clk = 0;
         #5;
         clk = 1;
         #5;
      end
```

Clock component

903

```
S9    // a signal generator
      always
      begin
         #10;
         sigin = $random;
      end
```

Test-bench component (Driver)

904

```
      // initialization
      initial
      begin
S10      reset = 0;
         sigin = 0;
         #1;
S11      reset = 1;
         #5;
S12      $monitor($time, "%b, %b", sigin, sigout);
         #1000 $finish;
      end
      end module
```

Test-bench component (initialization)

905

Test-bench component (monitor)

906

FIG. 28

SIGNAL NETWORK ANALYSIS

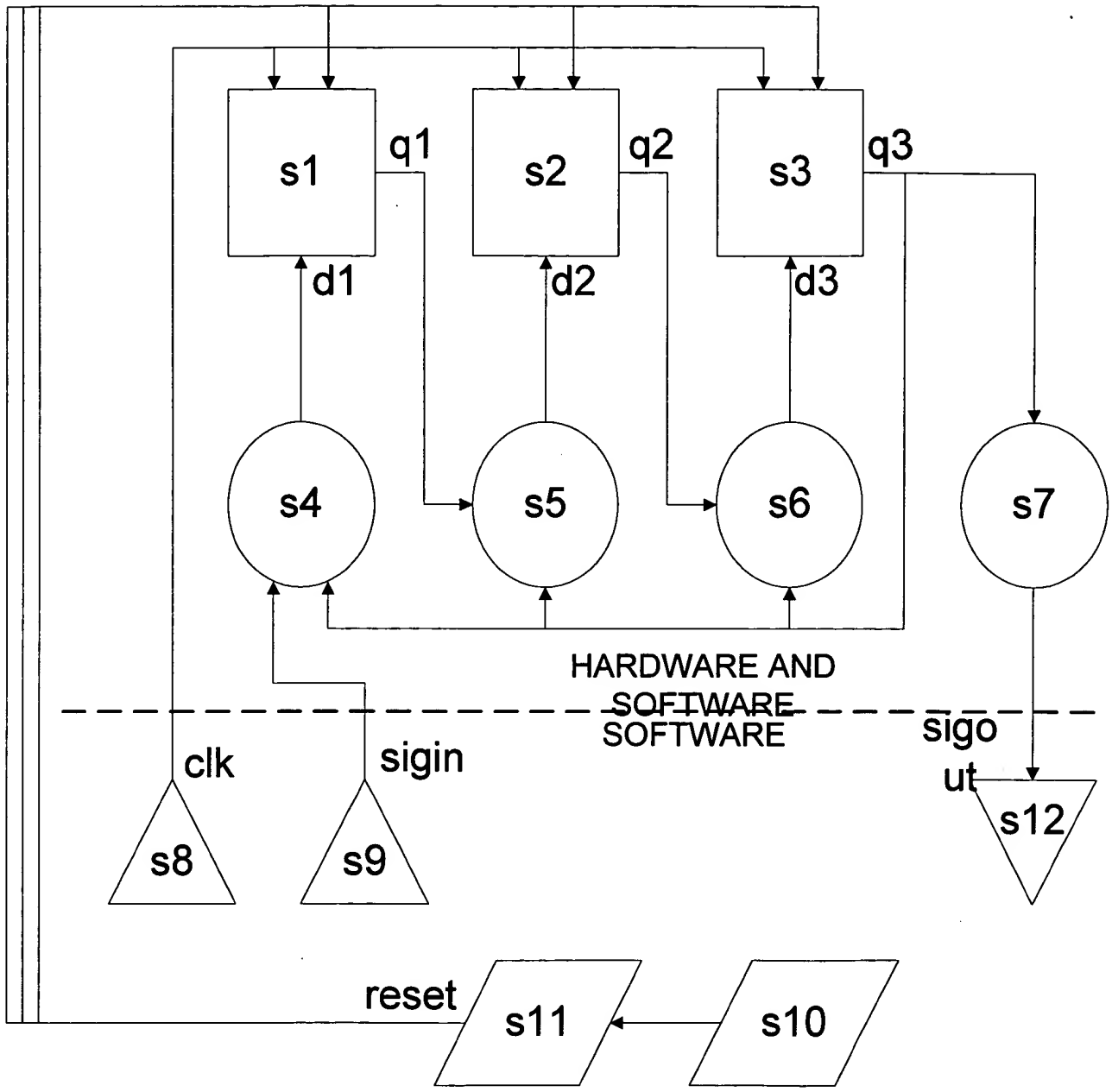


FIG 29

SOFTWARE/HARDWARE PARTITION RESULT

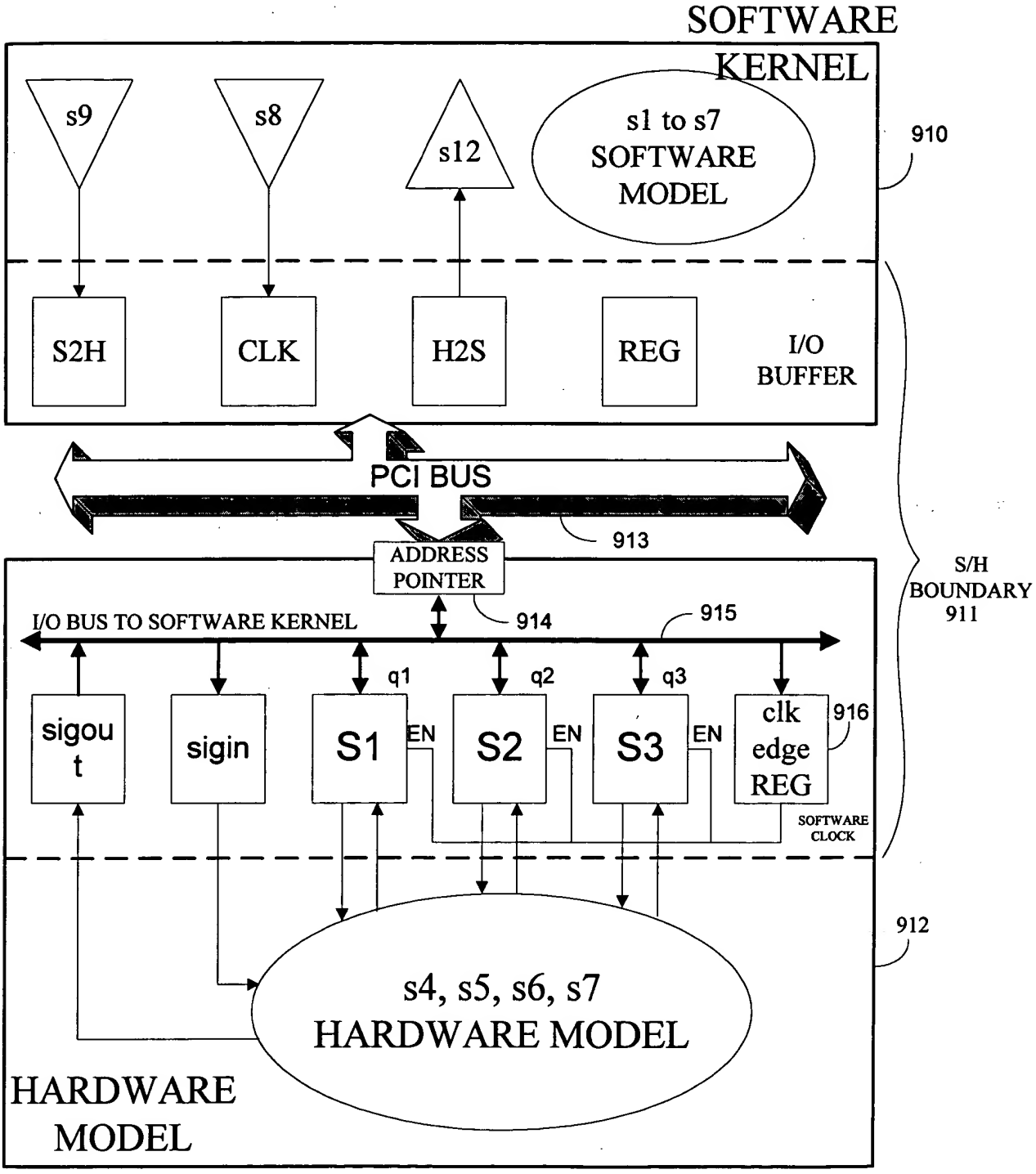


FIG 30

HARDWARE MODEL

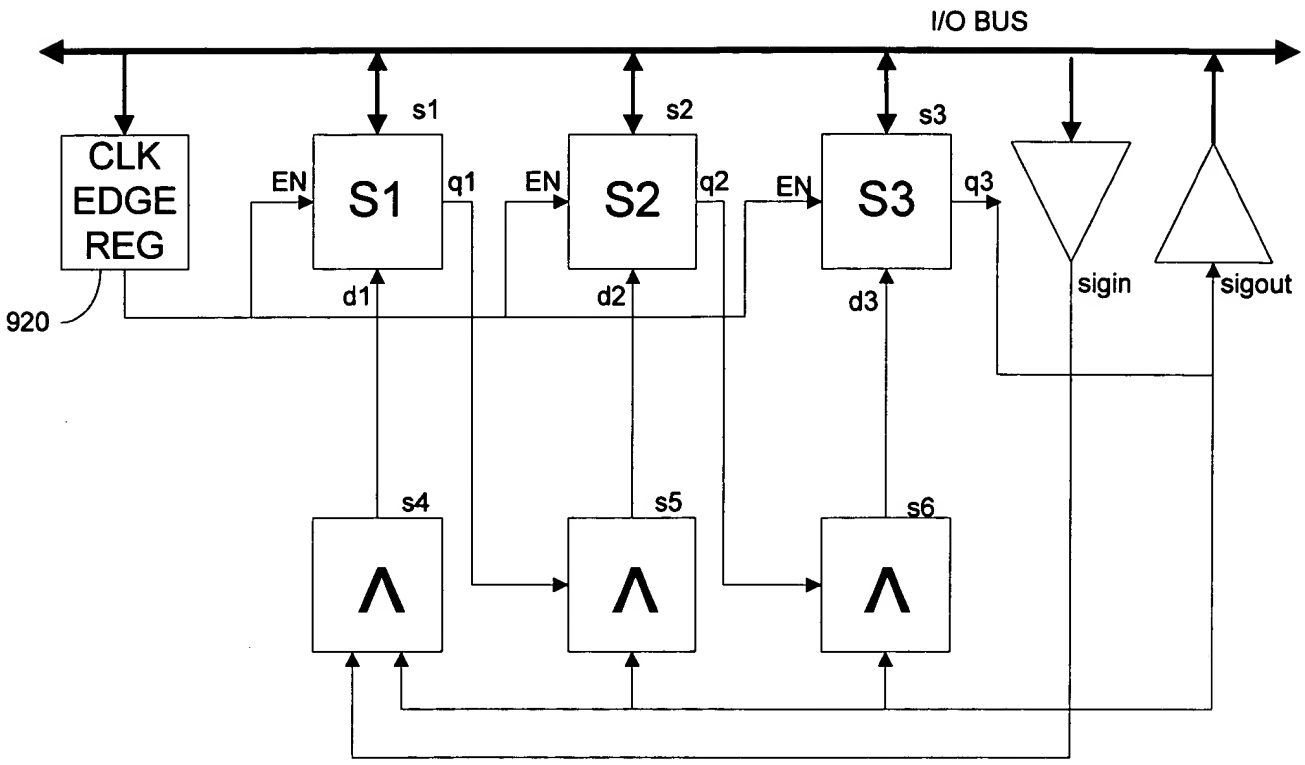
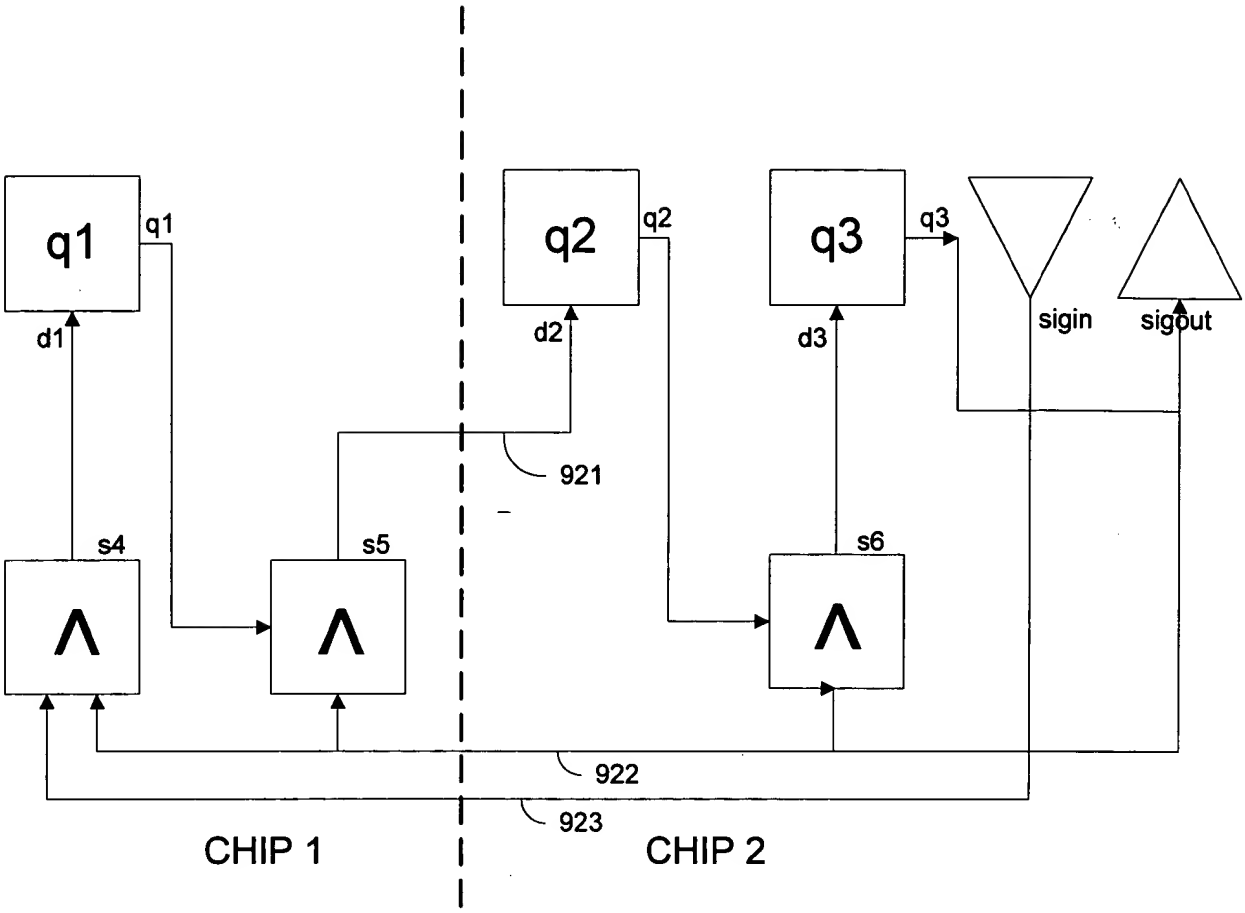


FIG. 31

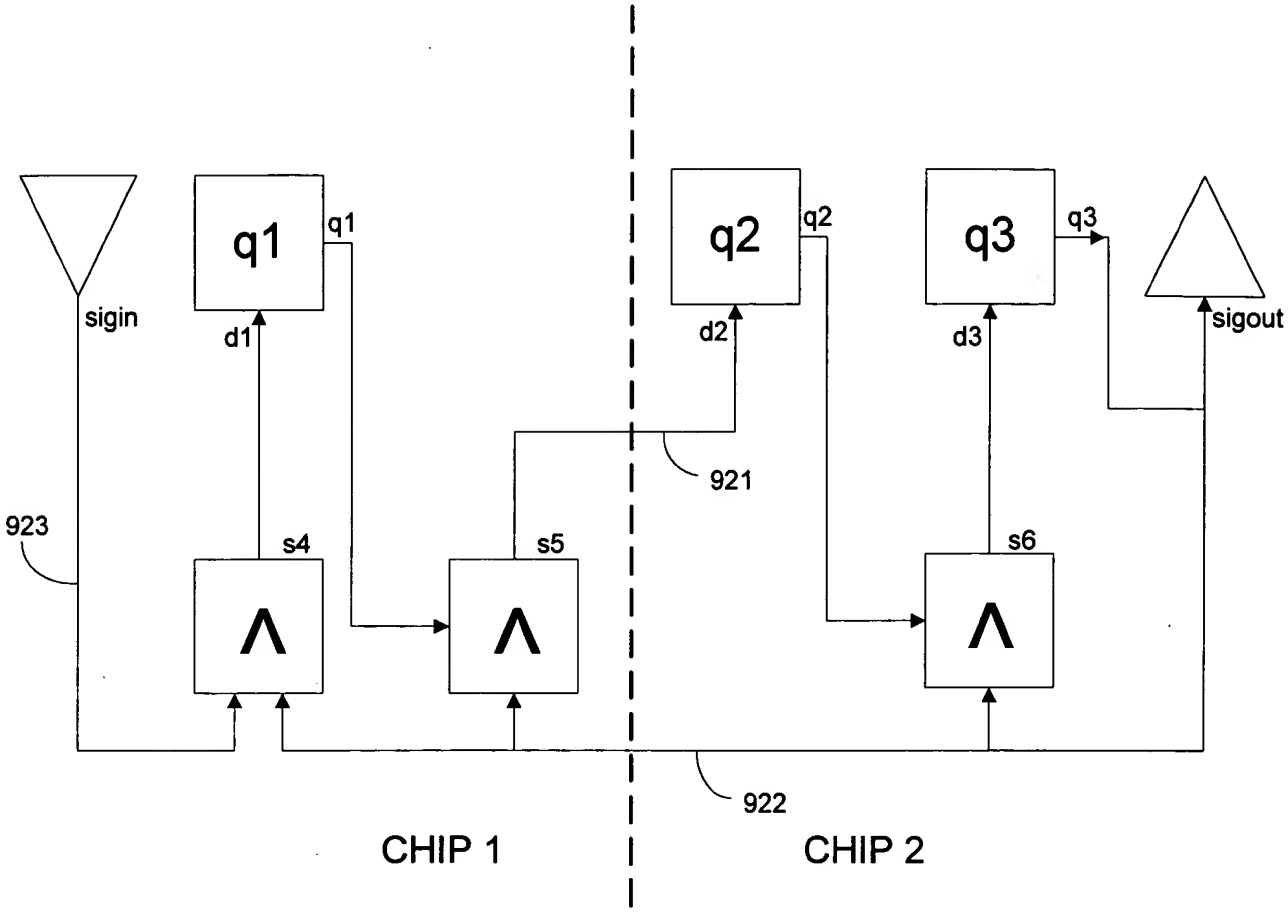
PARTITION RESULT #1



(IGNORE I/O AND CLOCK EDGE REGISTER)

FIG. 32

PARTITION RESULT
#2



(IGNORE I/O AND CLOCK EDGE REGISTER)

FIG. 33

LOGIC PATCHING

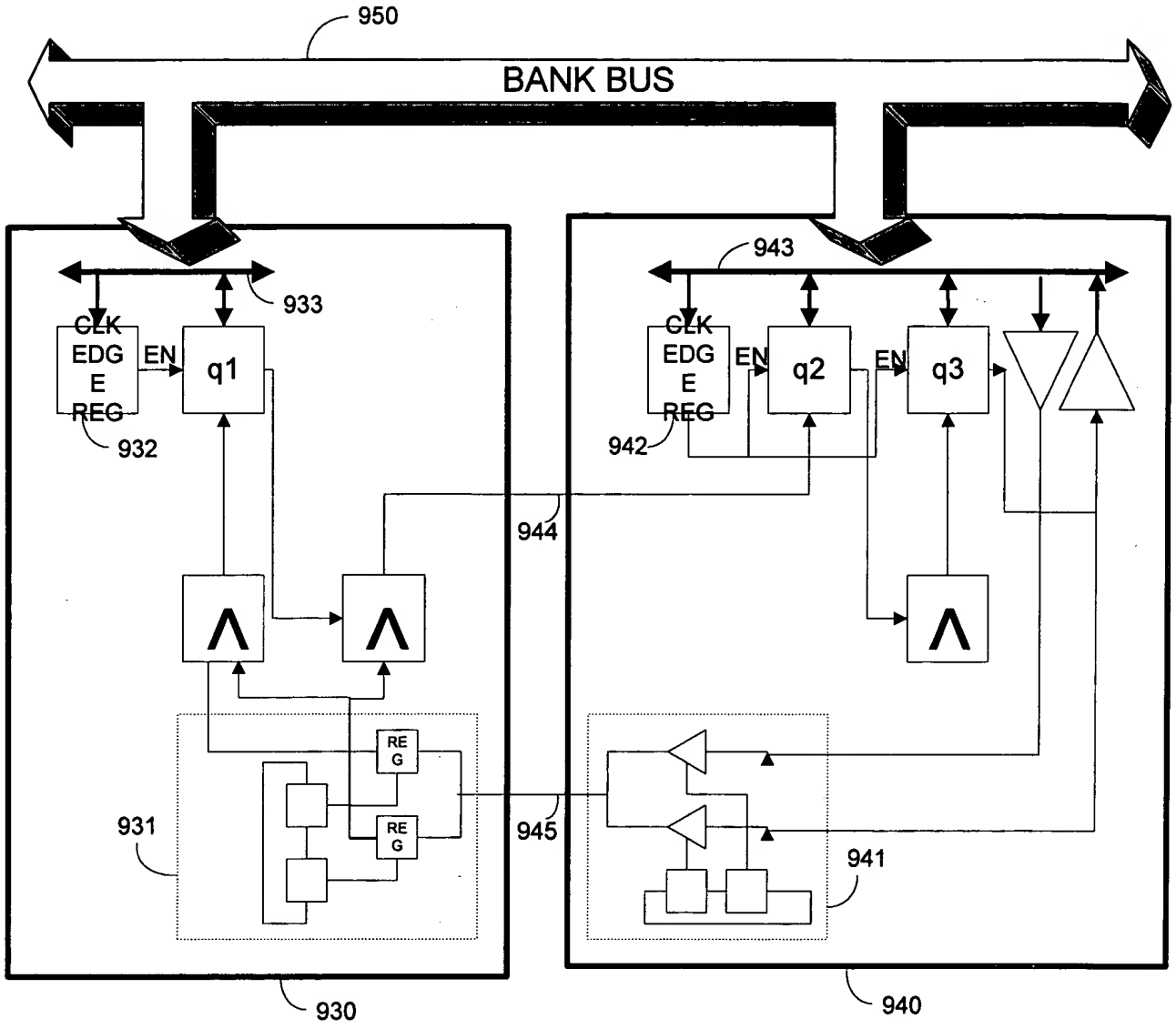
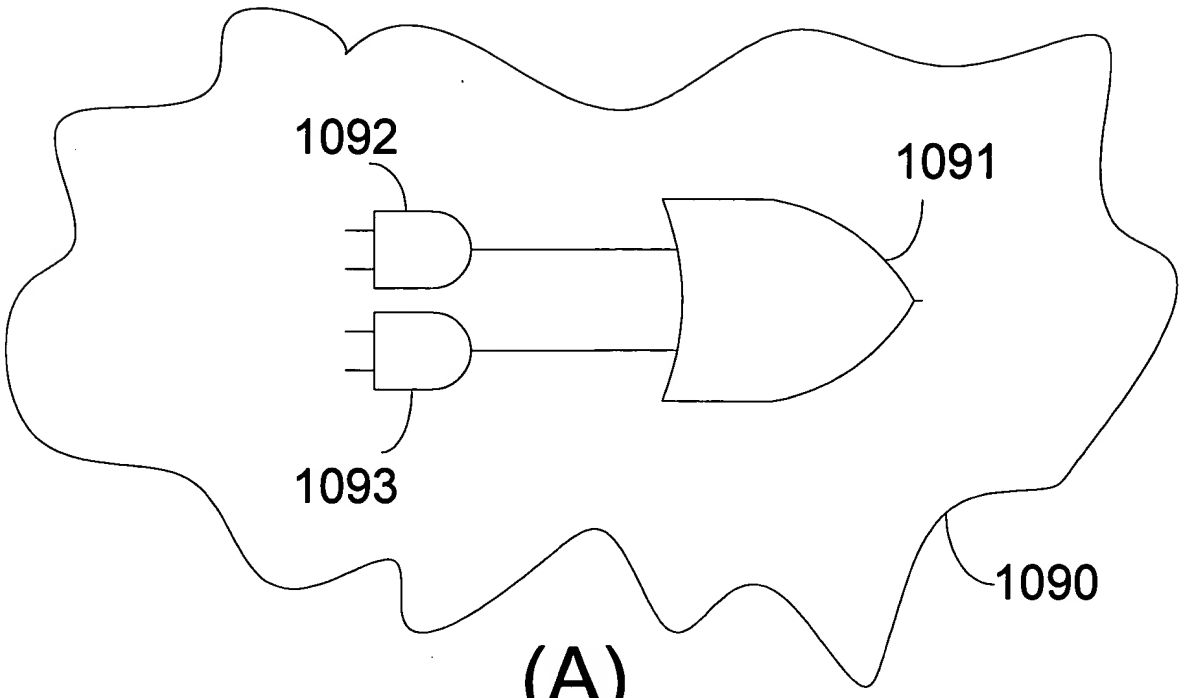
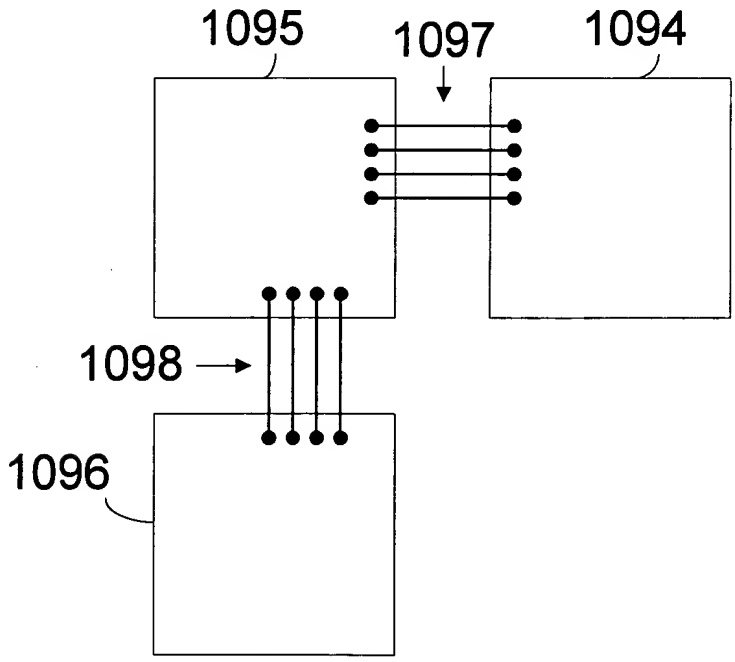


FIG. 34

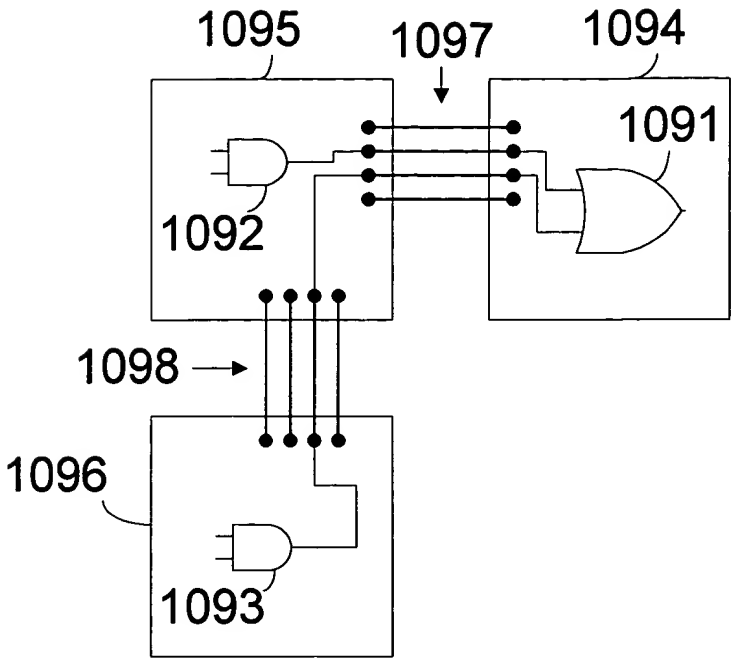


(A)

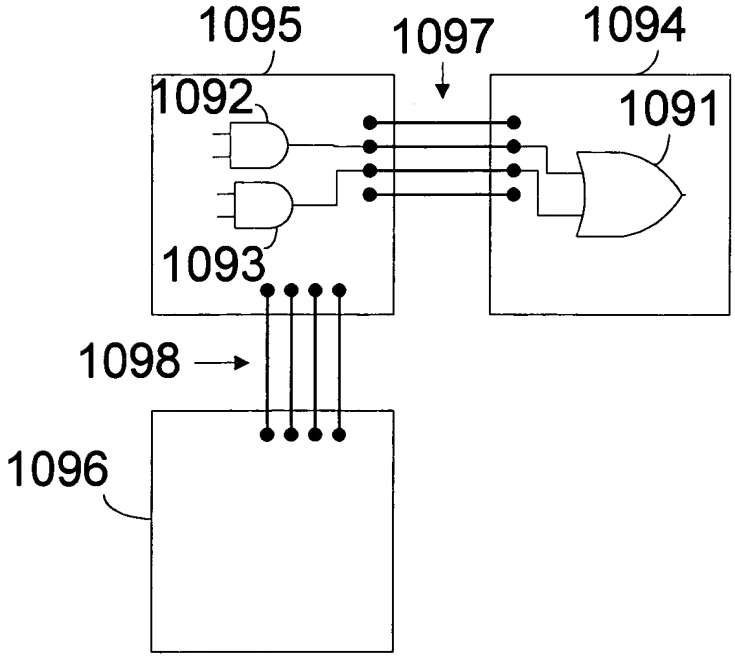


(B)

FIG. 35



(C)

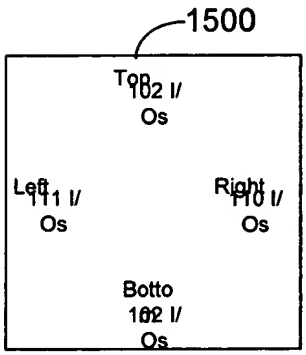


(D)

FIG 35

I/O PIN OVERVIEW OF FPGA LOGIC DEVICE

FPGA : 10K130V, 10K250V with 599-pin PGA package



425 Interconnect I/O pins

45 Dedicated I/O pins:

GCLK, BUS[31..0], F_RD,
FDATAXSFR, SHIFTWR,
SPACE[1..0], EXTEST,
DEV_OE, EQ_N,
DEV_CLRN

FIG. 36

FPGA INTERCONNECT BUSES

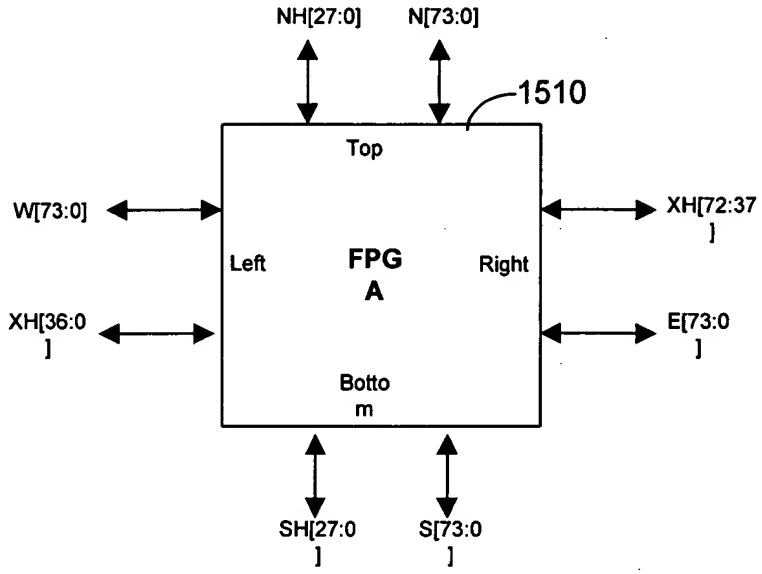


FIG 37

BOARD CONNECTION - SIDE VIEW

DUAL-BOARD CONFIGURATION

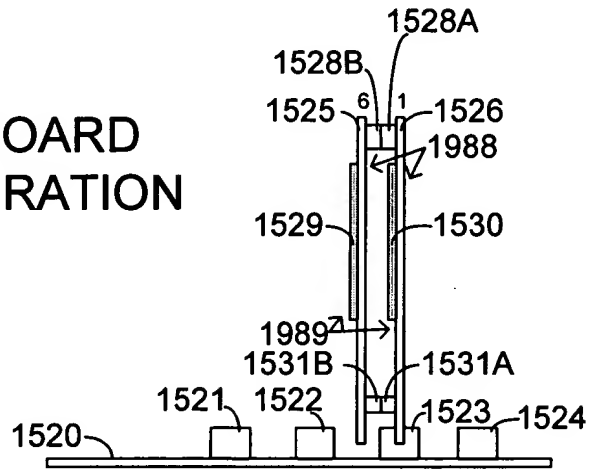


FIG. 38(A)

SIX BOARD CONFIGURATION

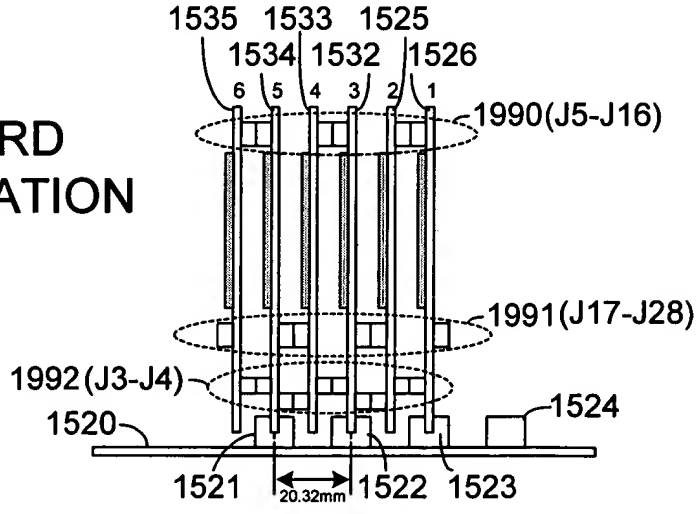


FIG 38(B)

SIX-BOARD CONFIGURATION
DIRECT-NEIGHBOR AND ONE-HOP FPGA ARRAY – X TORUS, Y MESH

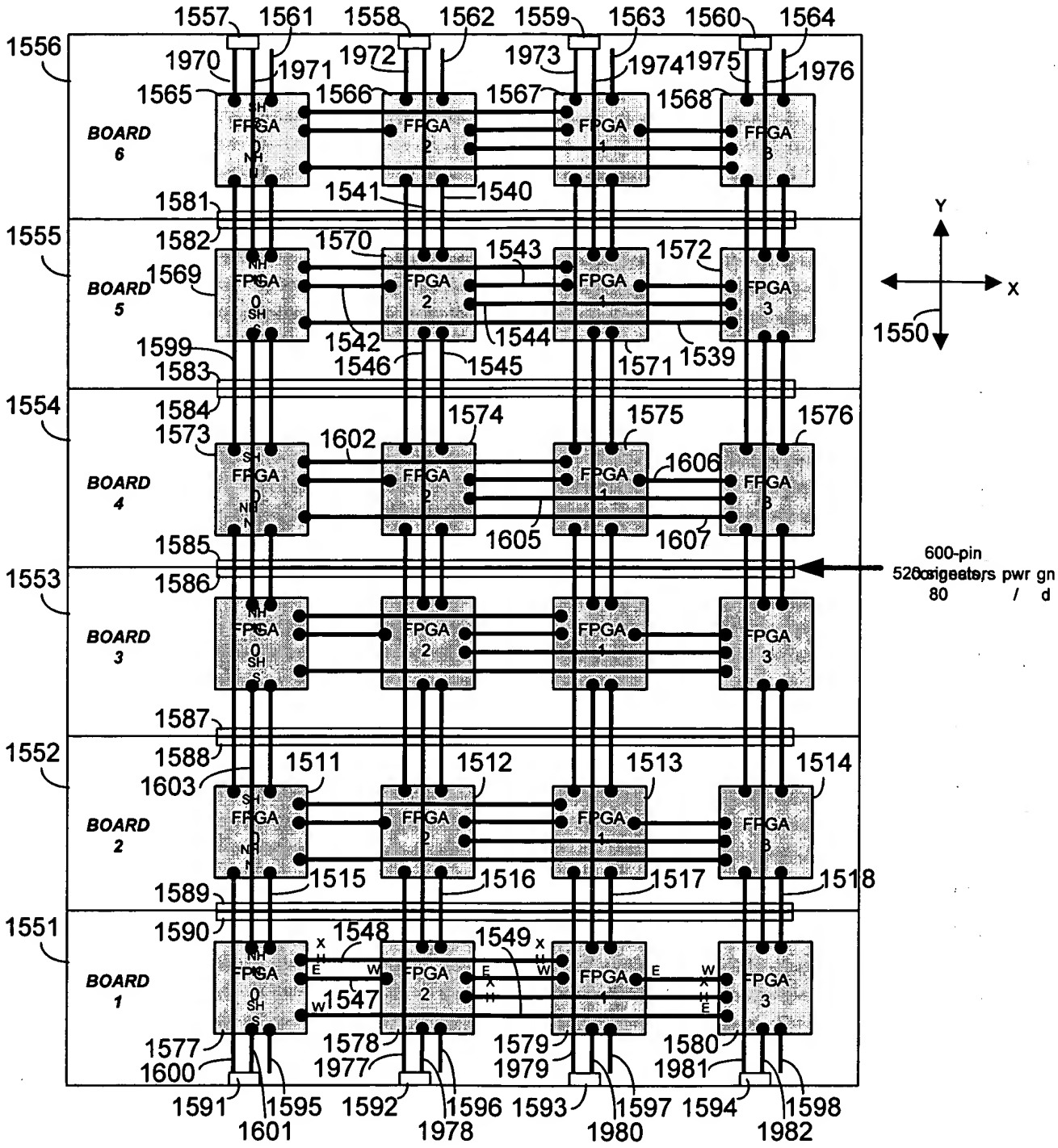


FIG 39

FPGA ARRAY CONNECTION BETWEEN BOARDS

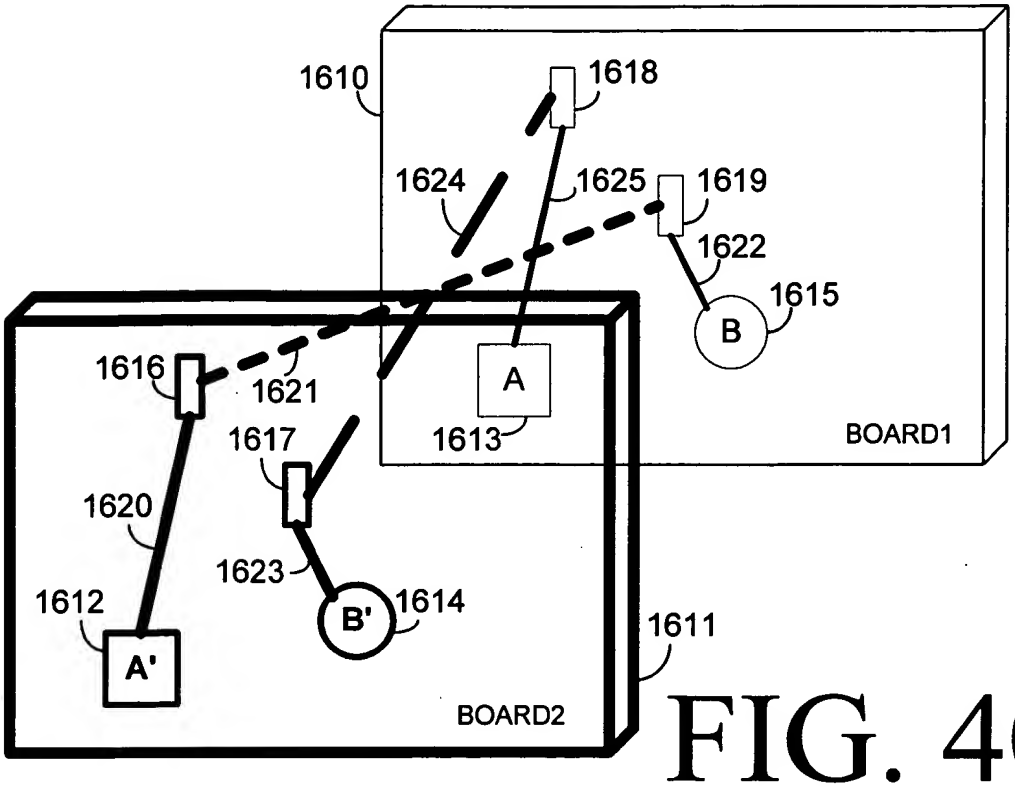


FIG. 40(A)

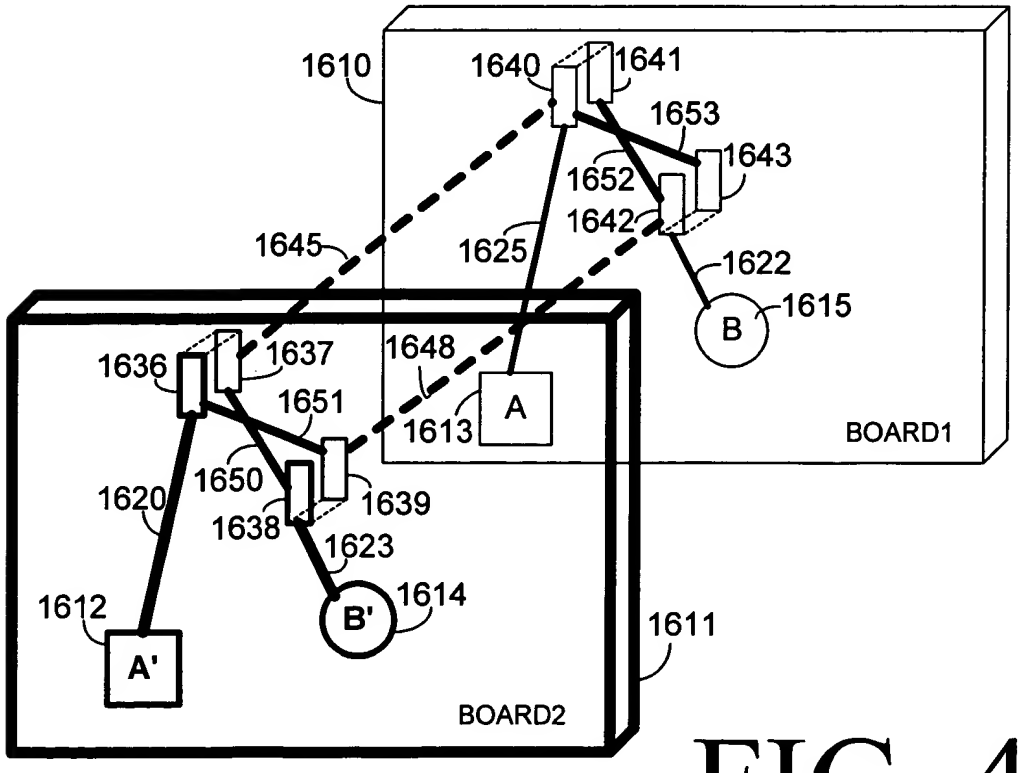


FIG 40(B)

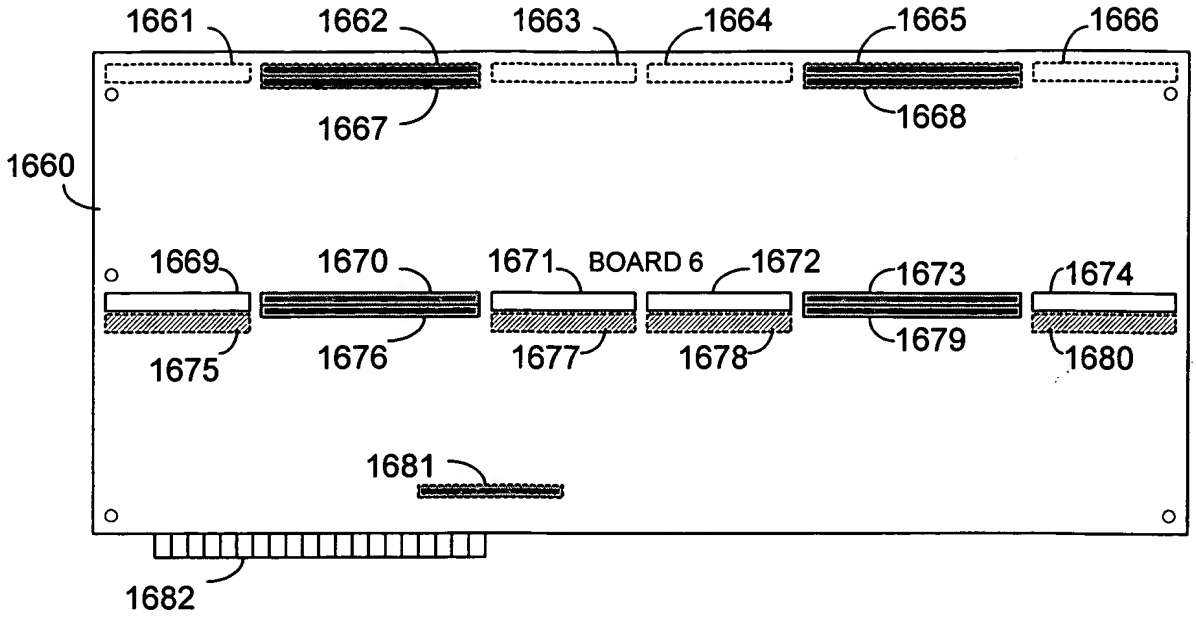


FIG. 41(A)

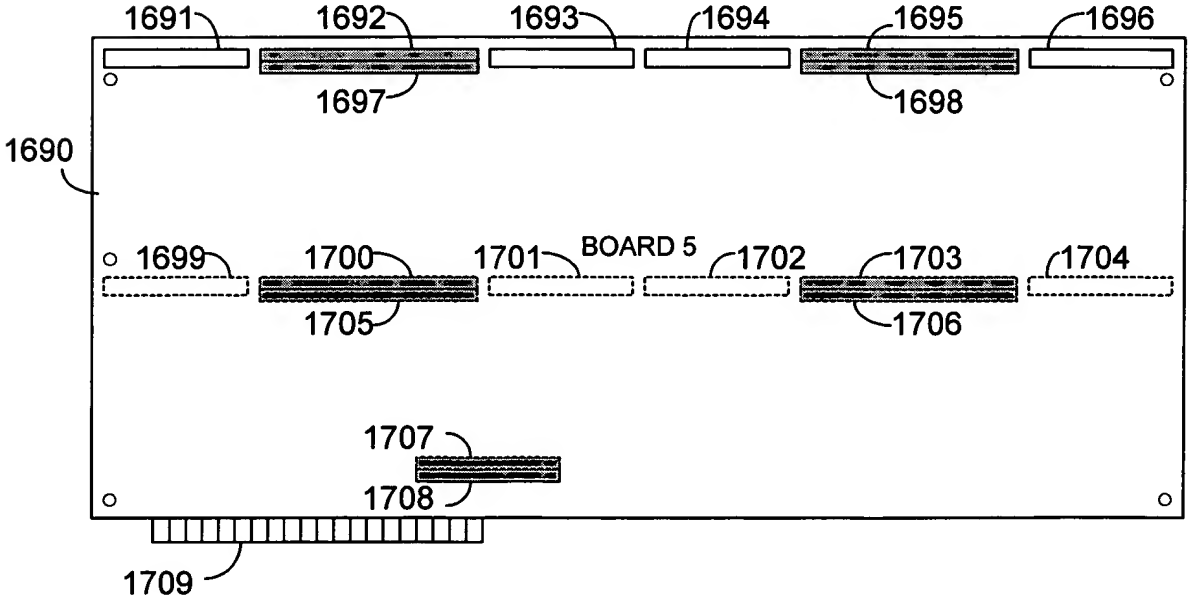


FIG 41(B)

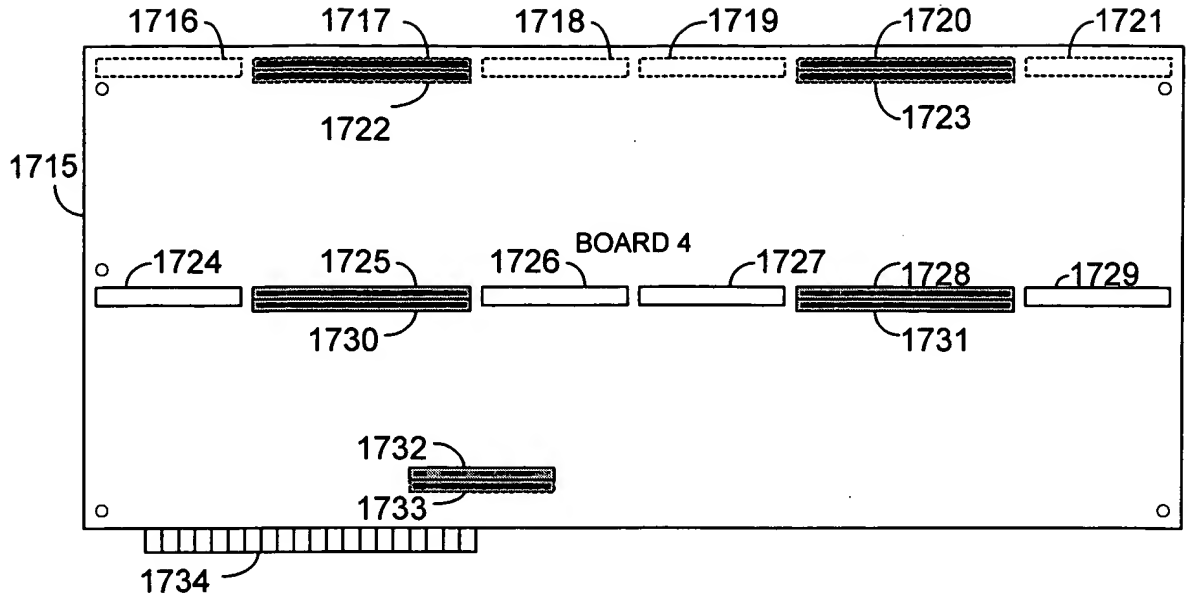


FIG. 41(C)

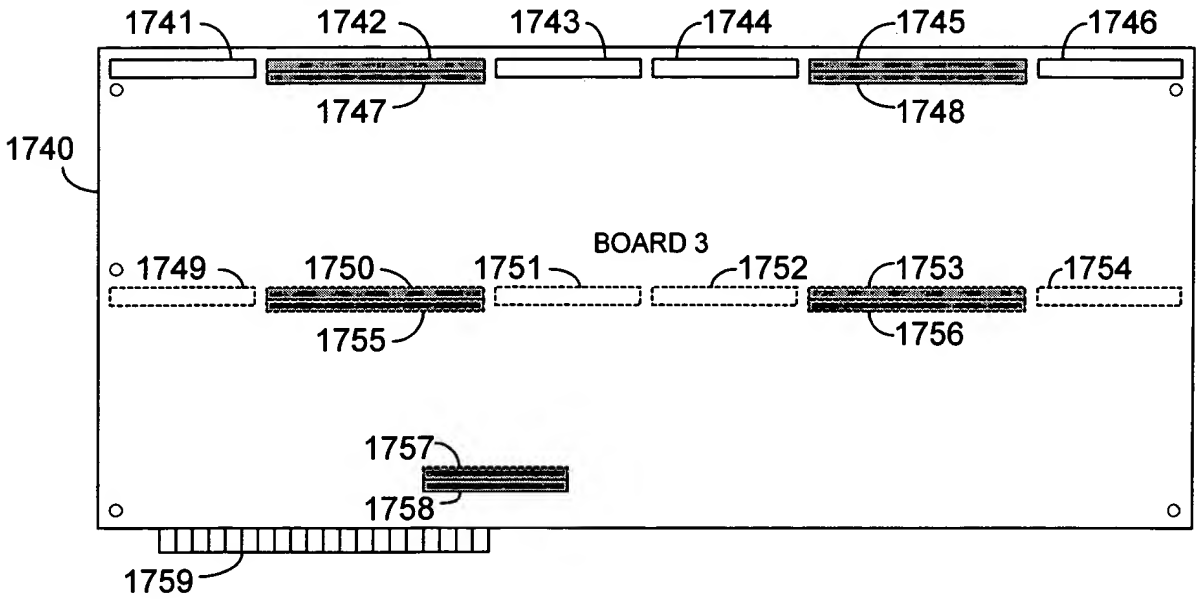


FIG. 41(D)

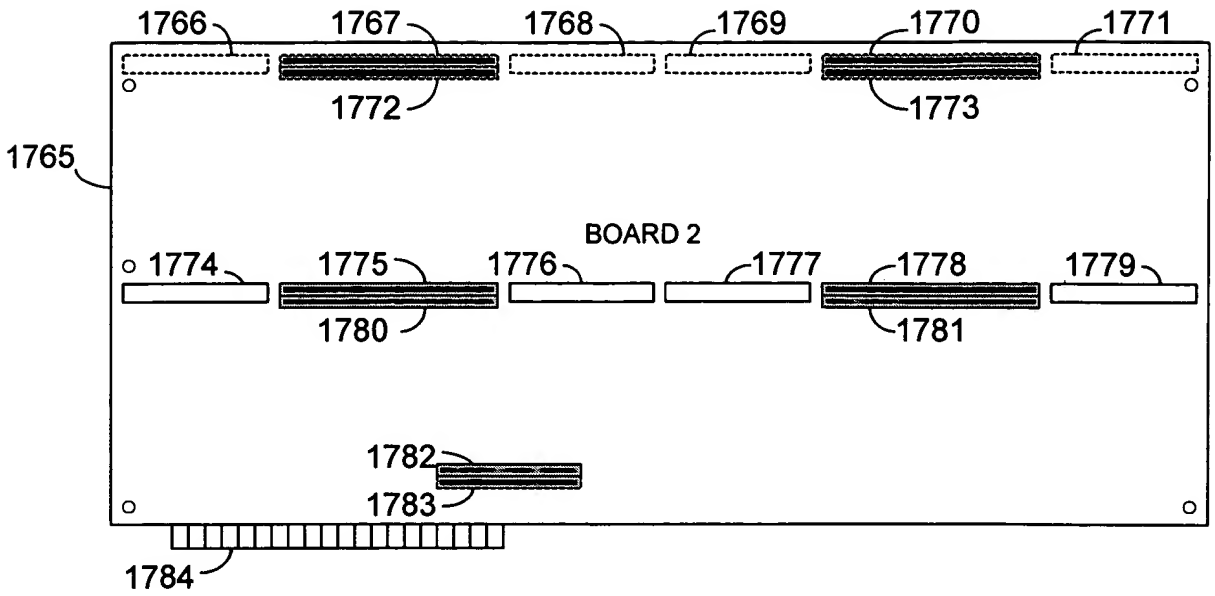


FIG. 41(E)

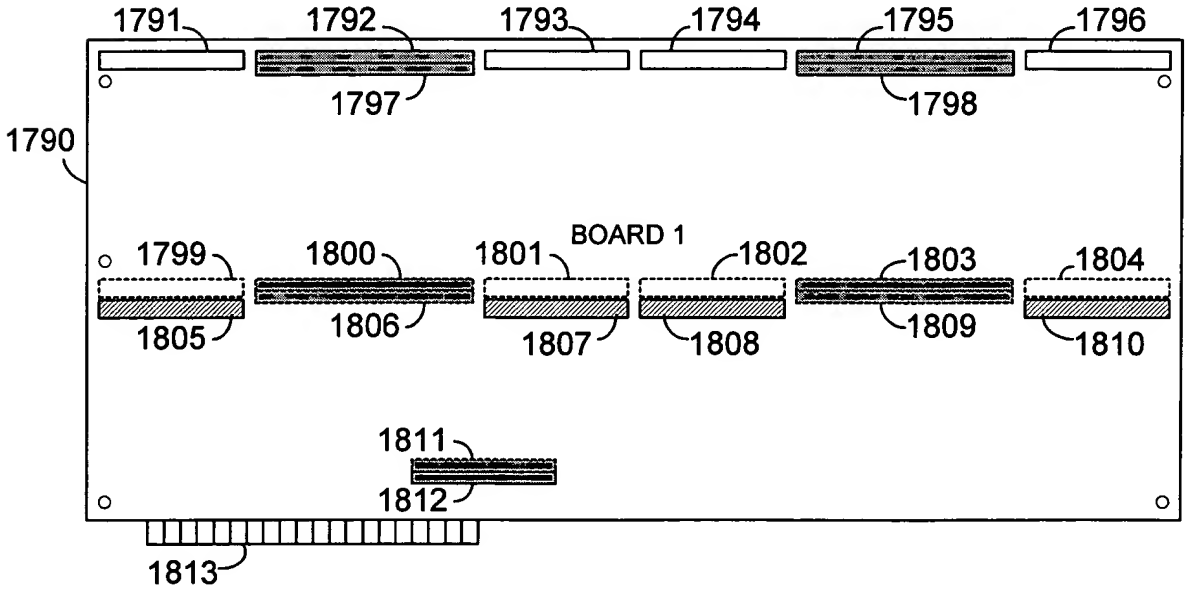


FIG 41(F)

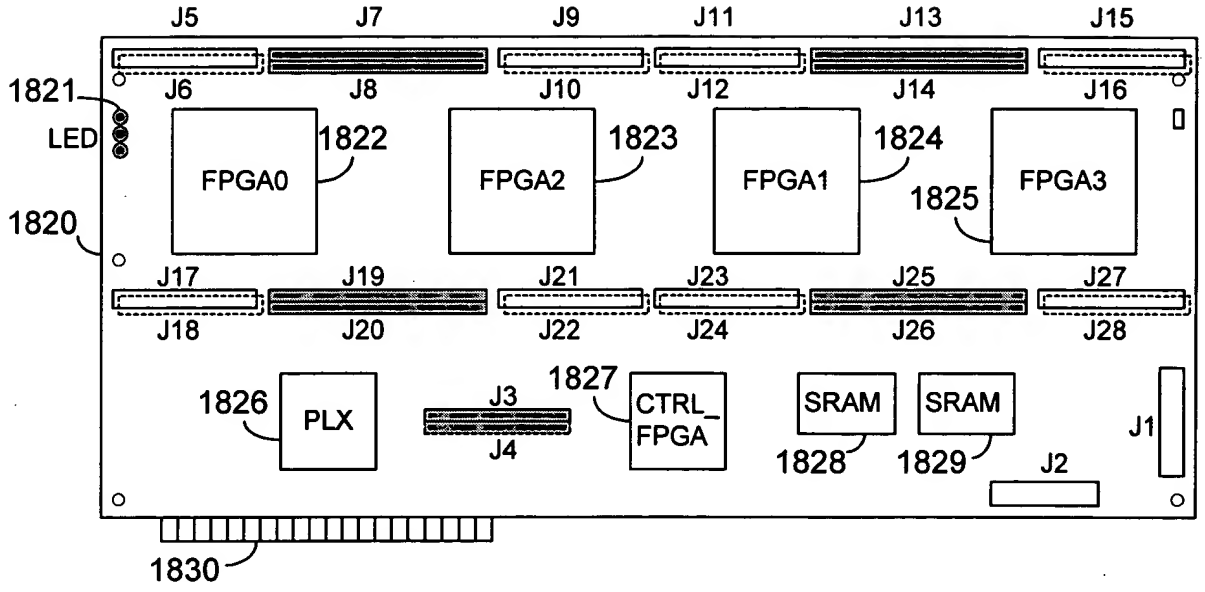

J5
J7
J9
J11
J13
J15
J6
J8
J10
J12
J14
J16
1821
LED
1820
FPGA0
1822
FPGA2
1823
FPGA1
1824
FPGA3
1825
J17
J19
J21
J23
J25
J27
J18
J20
J22
J24
J26
J28
1826
PLX
J3
J4
1827
CTRL_
FPGA
SRAM
SRAM
1828
1829
J1
J2
1830

FIG. 42

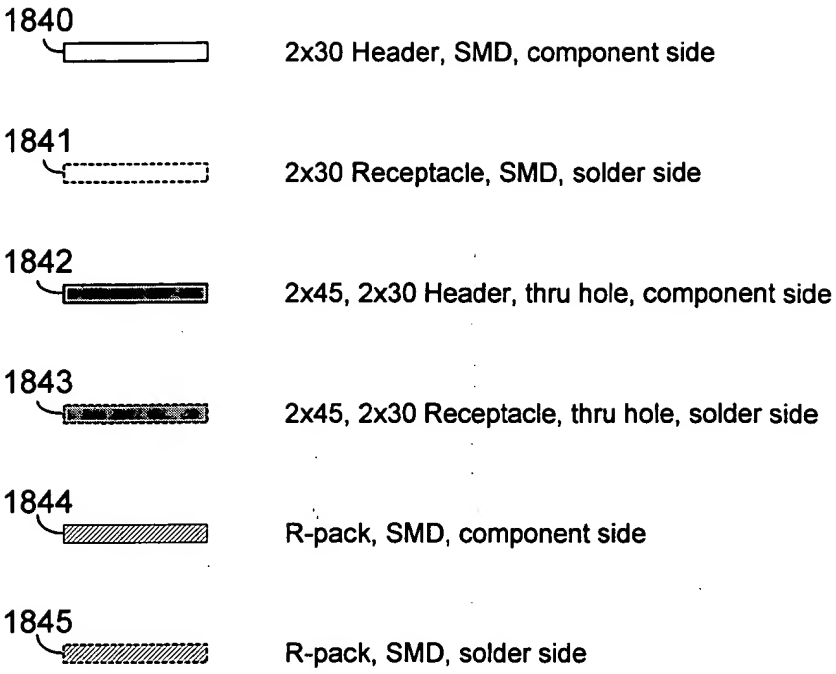

1840
2x30 Header, SMD, component side
1841
2x30 Receptacle, SMD, solder side
1842
2x45, 2x30 Header, thru hole, component side
1843
2x45, 2x30 Receptacle, thru hole, solder side
1844
R-pack, SMD, component side
1845
R-pack, SMD, solder side

FIG. 43

TWO-BOARD CONFIGURATION
DIRECT-NEIGHBOR AND ONE-HOP FPGA ARRAY – X TORUS, Y MESH

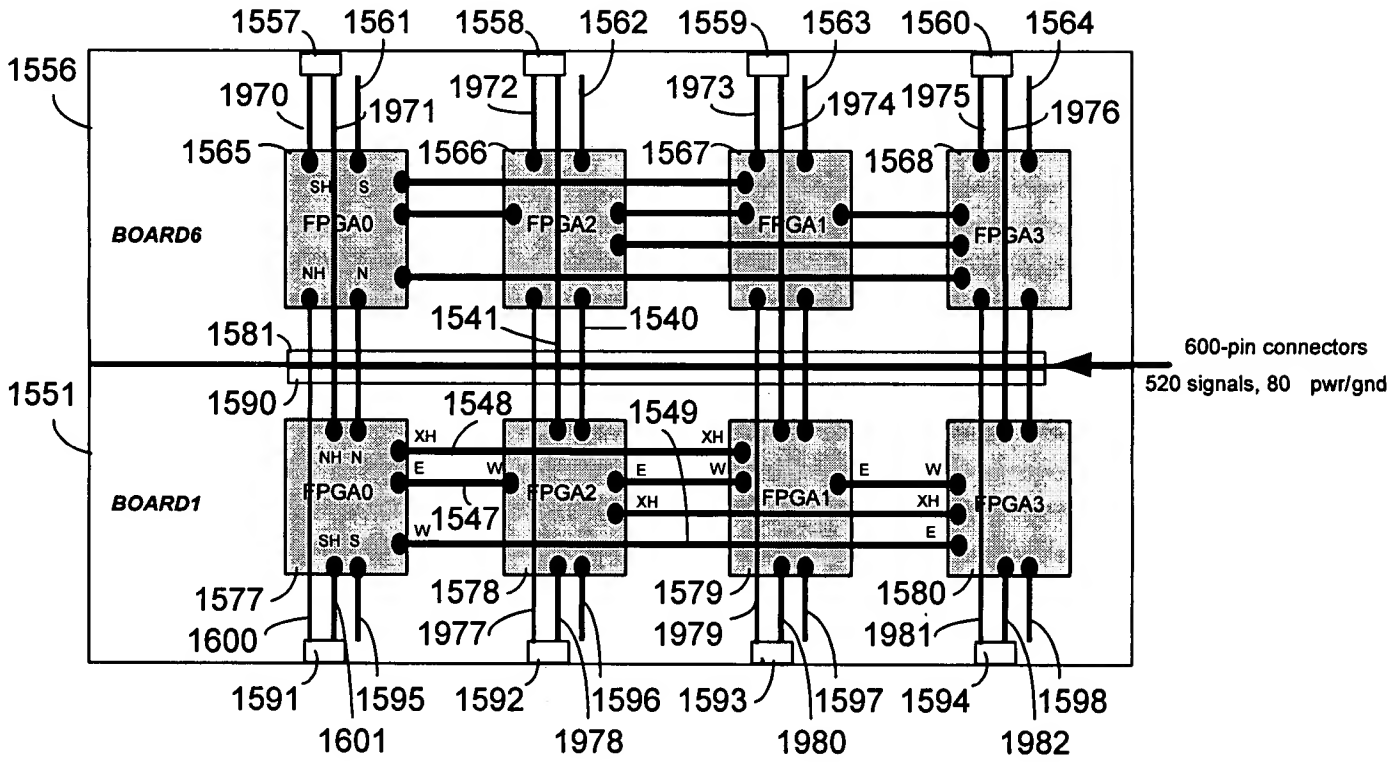


FIG. 44

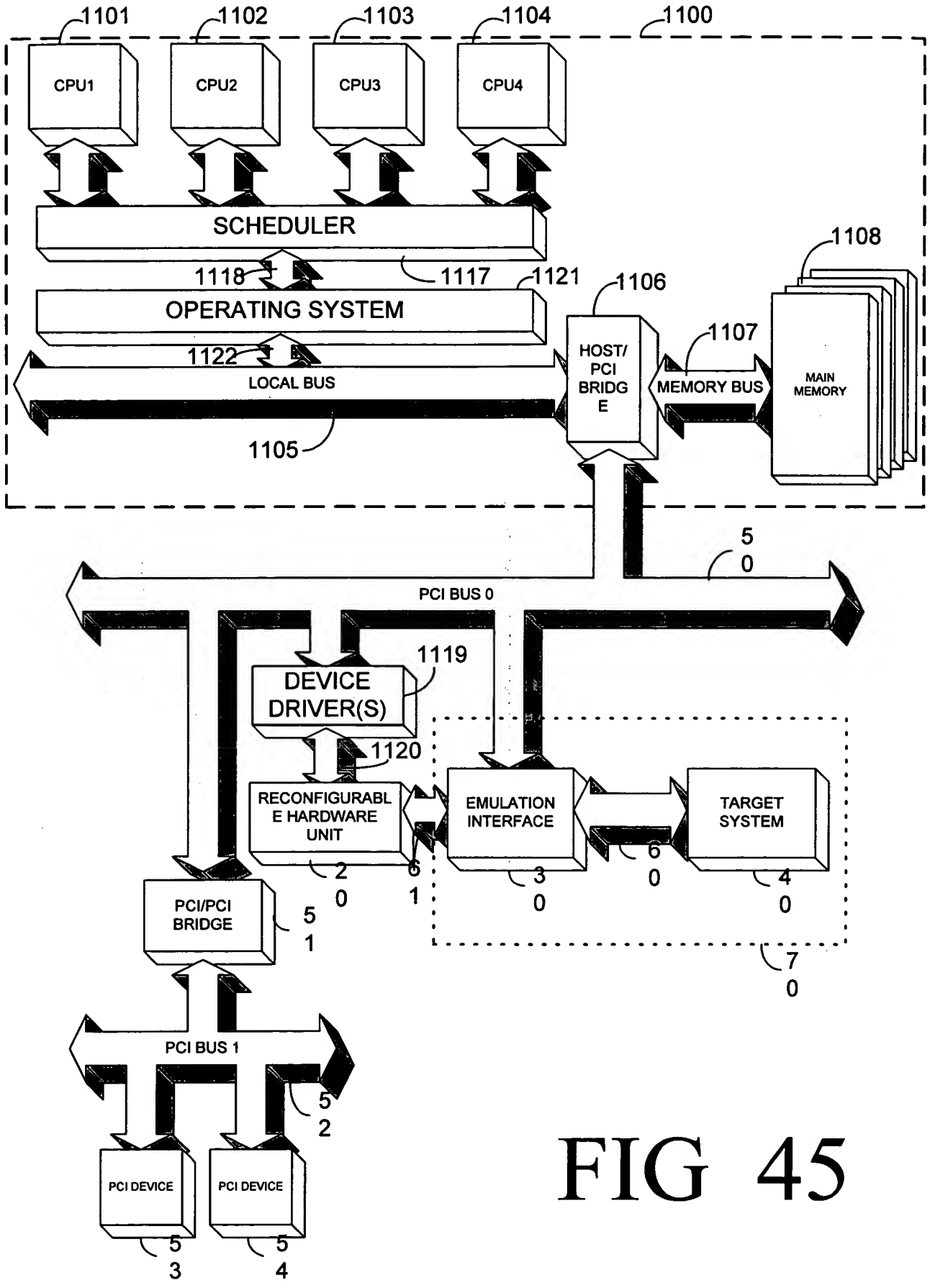


FIG 45

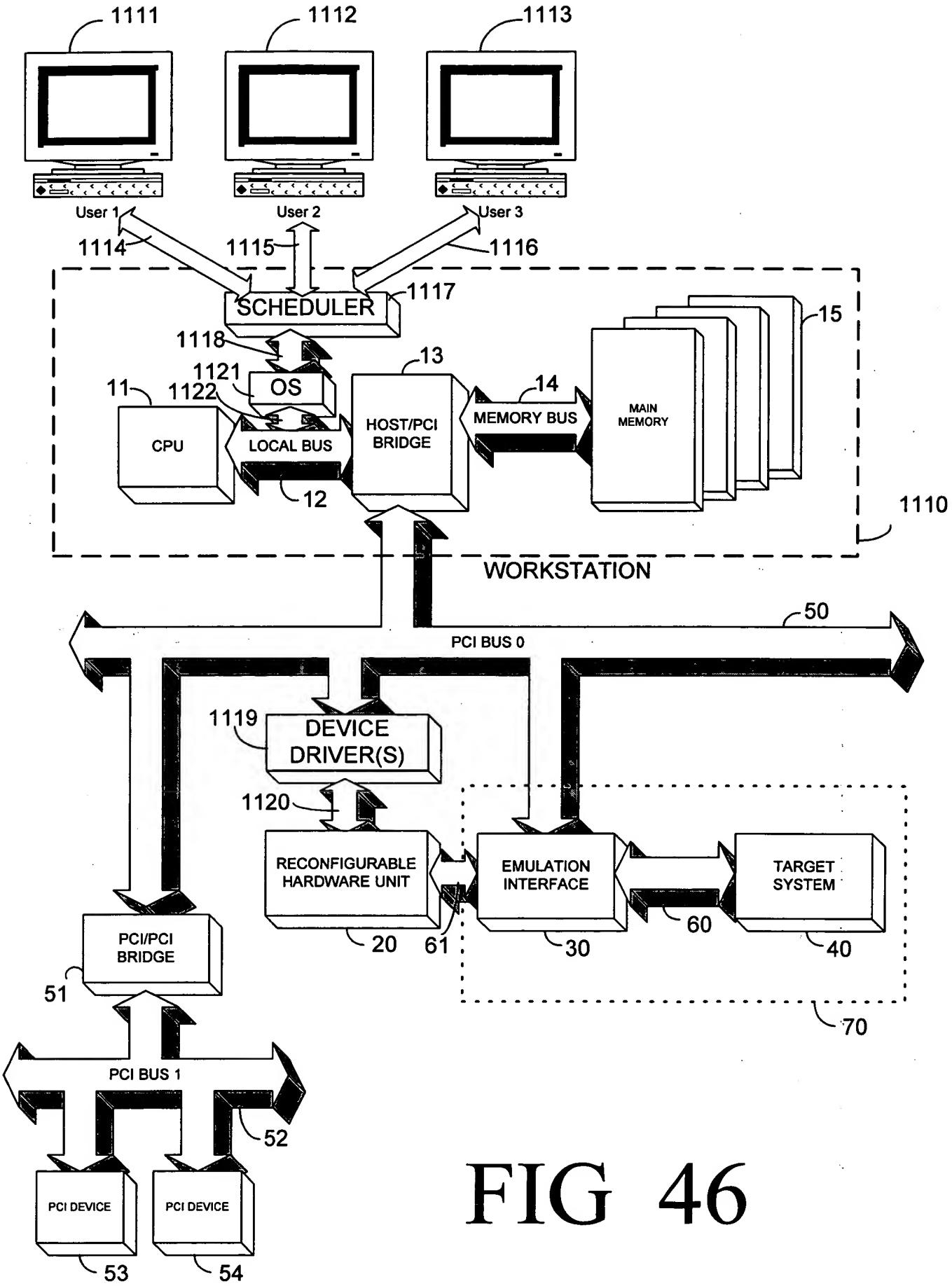

1111
1112
1113
User 1
User 2
User 3
1114
1115
1116
SCHEDULER
1117
1118
1121
OS
11
1122
CPU
LOCAL BUS
HOST/PCI BRIDGE
13
14
MEMORY BUS
MAIN MEMORY
15
12
1110
WORKSTATION
50
PCI BUS 0
1119
DEVICE DRIVER(S)
1120
RECONFIGURABLE HARDWARE UNIT
EMULATION INTERFACE
TARGET SYSTEM
20
61
30
60
40
70
PCI/PCI BRIDGE
51
PCI BUS 1
52
PCI DEVICE
PCI DEVICE
53
54

FIG 46

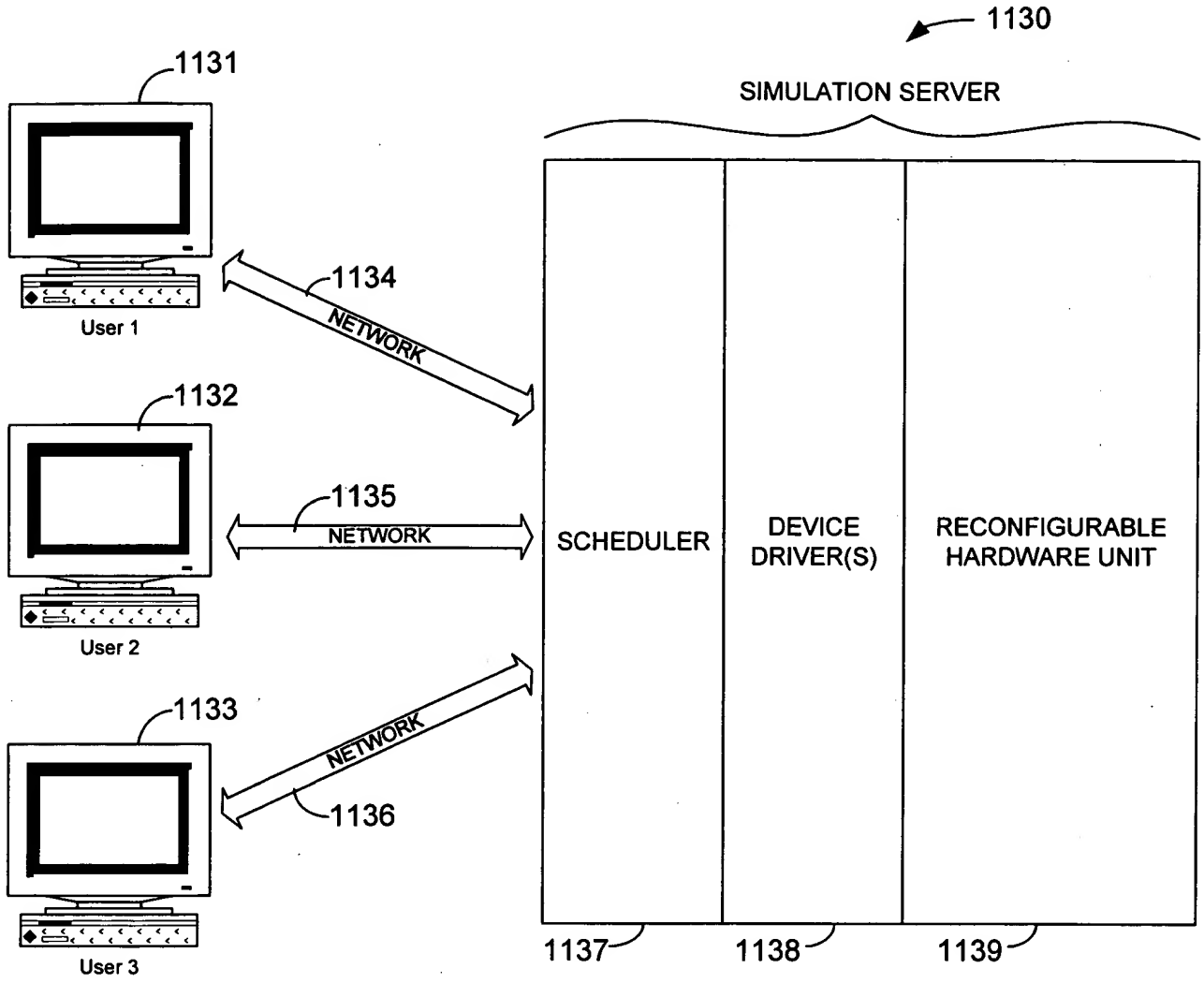

1130
SIMULATION SERVER
1131
User 1
1134
NETWORK
1132
User 2
1135
NETWORK
1133
NETWORK
1136
User 3
SCHEDULER
DEVICE DRIVER(S)
RECONFIGURABLE HARDWARE UNIT
1137
1138
1139

FIG. 47

SIMULATION SERVER ARCHITECTURE

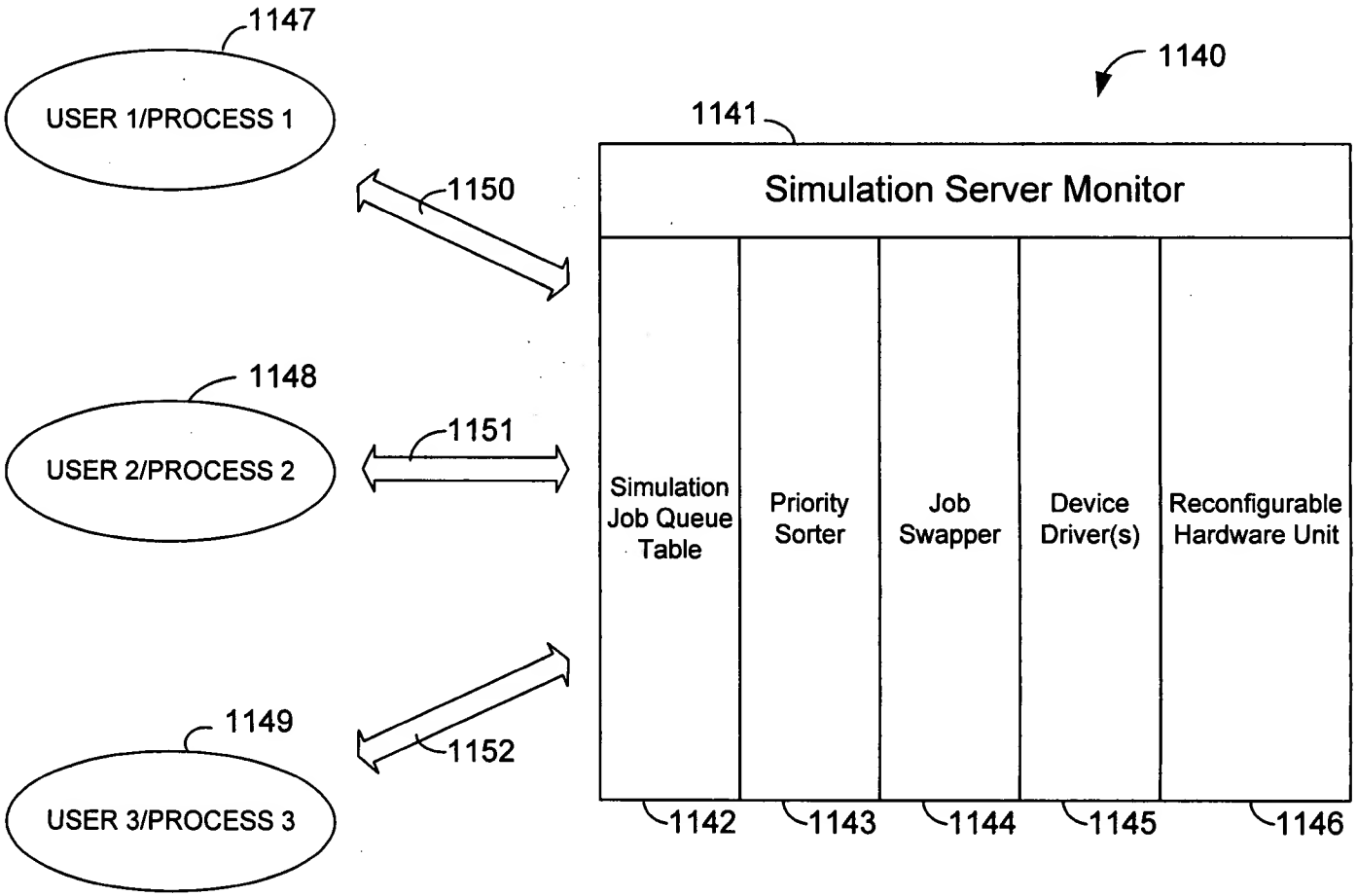


FIG 48

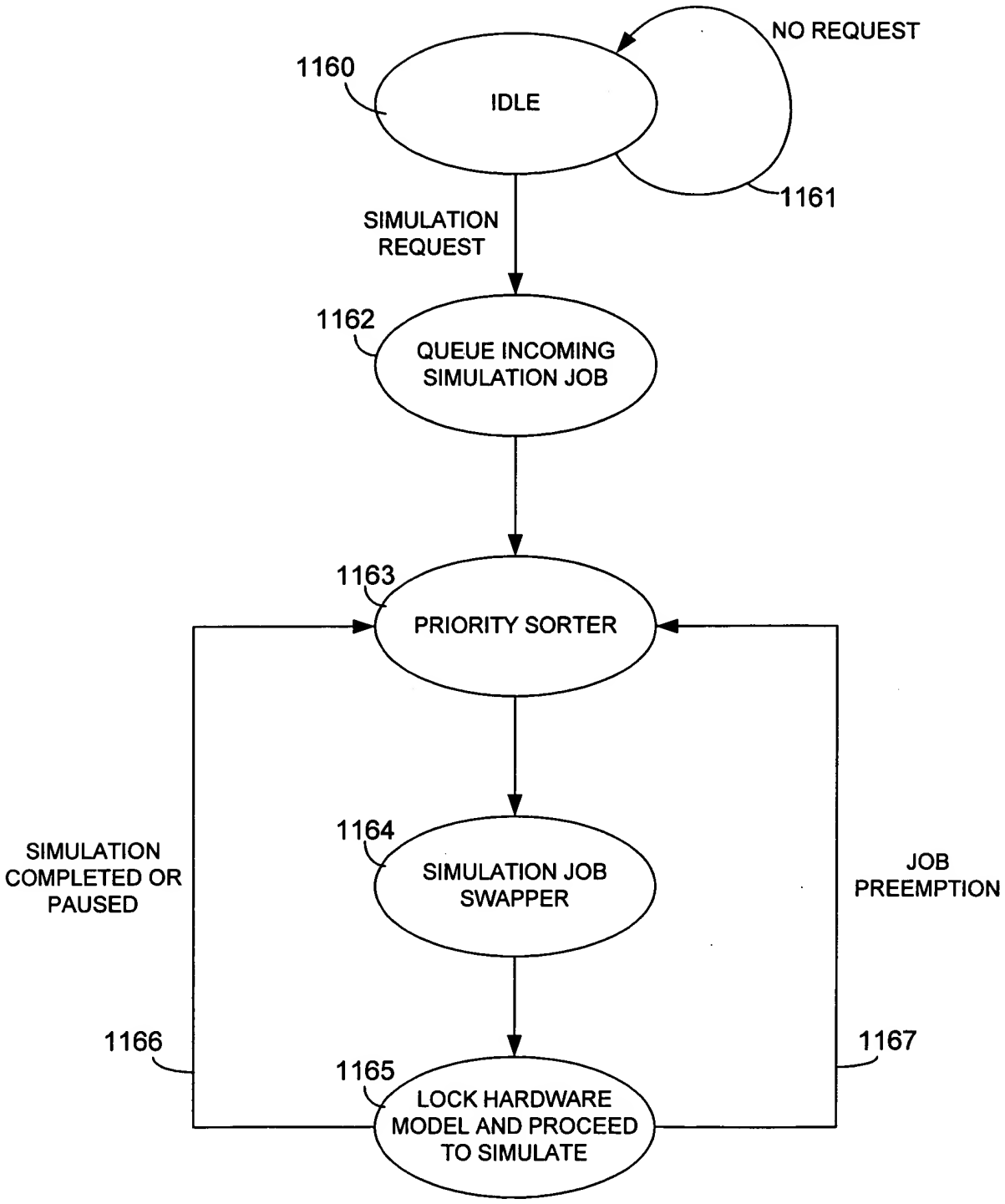


FIG. 49

JOB SWAPPER

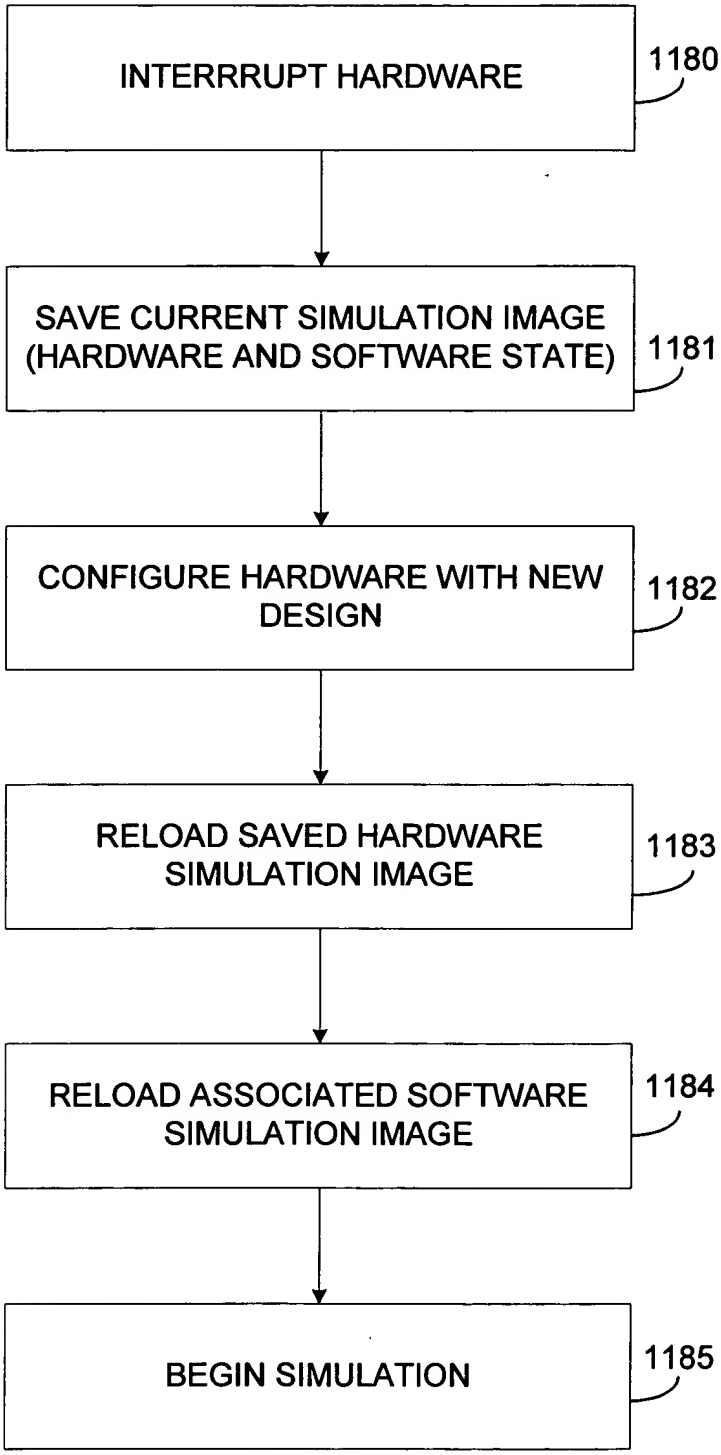


FIG 50

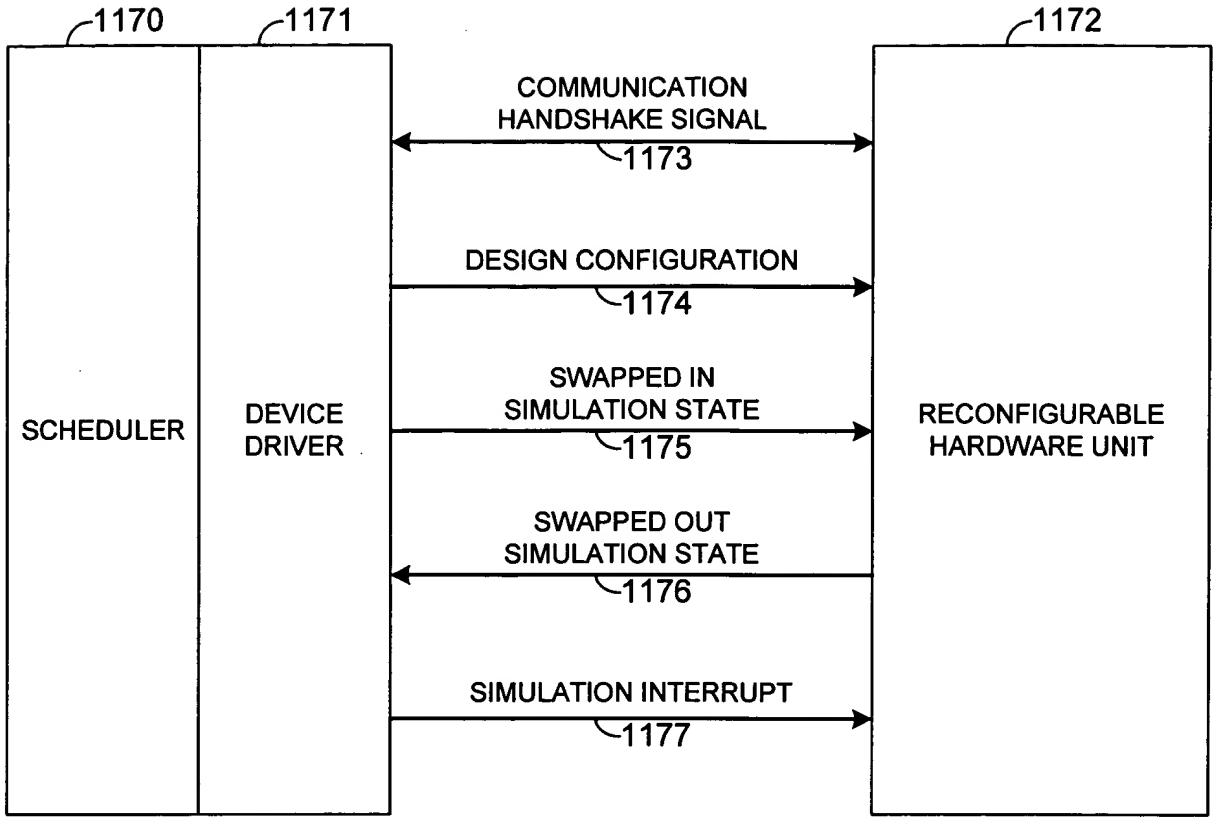

1170
1171
1172
SCHEDULER
DEVICE DRIVER
RECONFIGURABLE HARDWARE UNIT
COMMUNICATION HANDSHAKE SIGNAL
1173
DESIGN CONFIGURATION
1174
SWAPPED IN SIMULATION STATE
1175
SWAPPED OUT SIMULATION STATE
1176
SIMULATION INTERRUPT
1177

FIG. 51

PRIORITY I { JOB A, JOB B }

PRIORITY II { JOB C, JOB D }

TIME-SHARED HARDWARE USAGE:

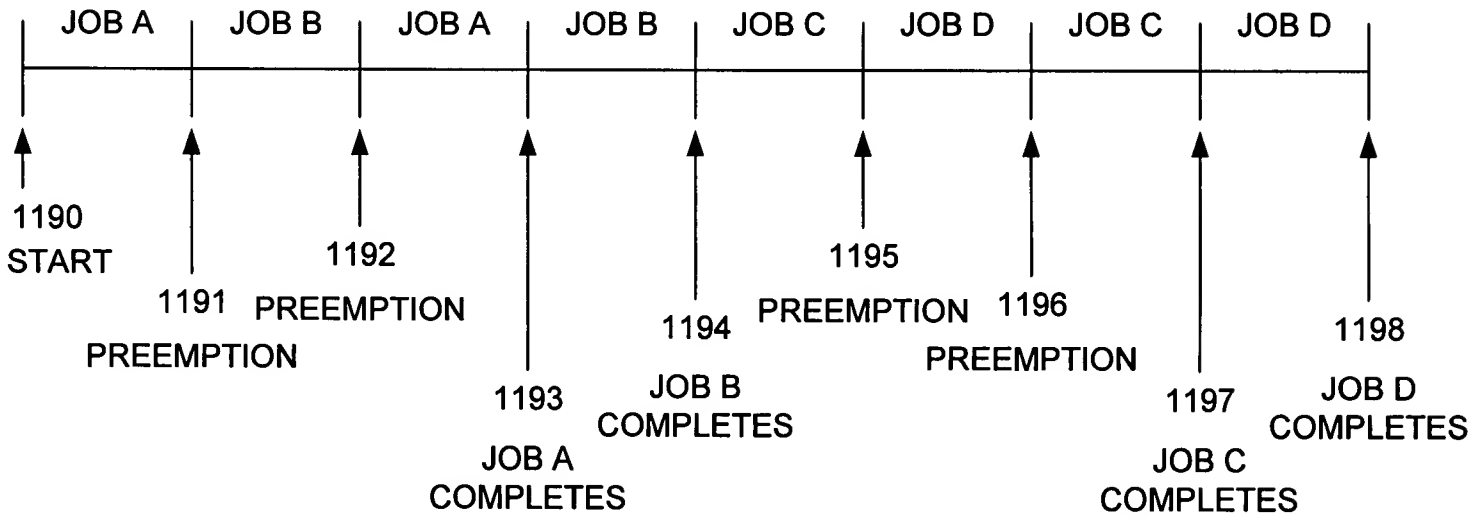


FIG 52

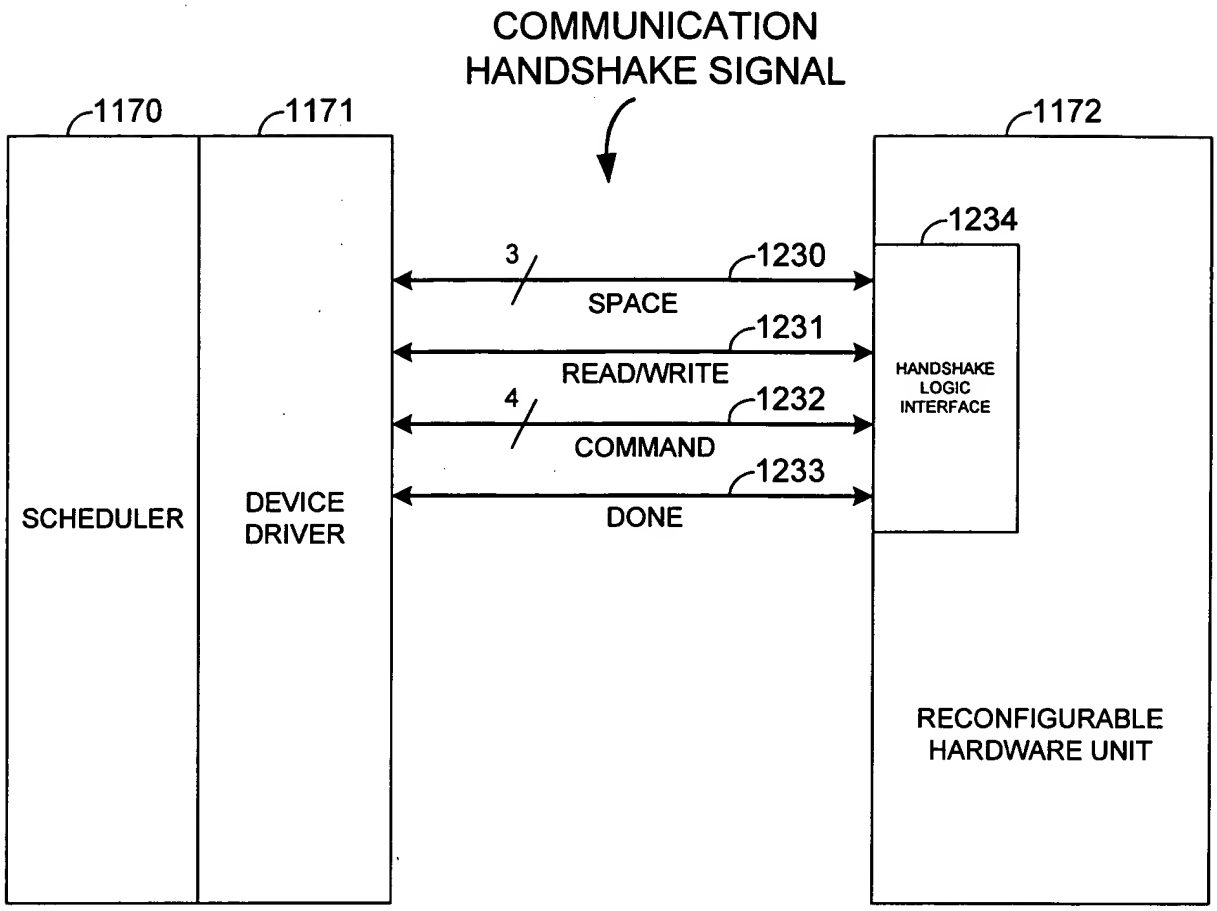

COMMUNICATION
HANDSHAKE SIGNAL
1170
1171
1172
1234
3
1230
SPACE
1231
READ/WRITE
HANDSHAKE
LOGIC
INTERFACE
4
1232
COMMAND
1233
DONE
SCHEDULER
DEVICE
DRIVER
RECONFIGURABLE
HARDWARE UNIT

FIG. 53

COMMUNICATION HANDSHAKE PROTOCOL

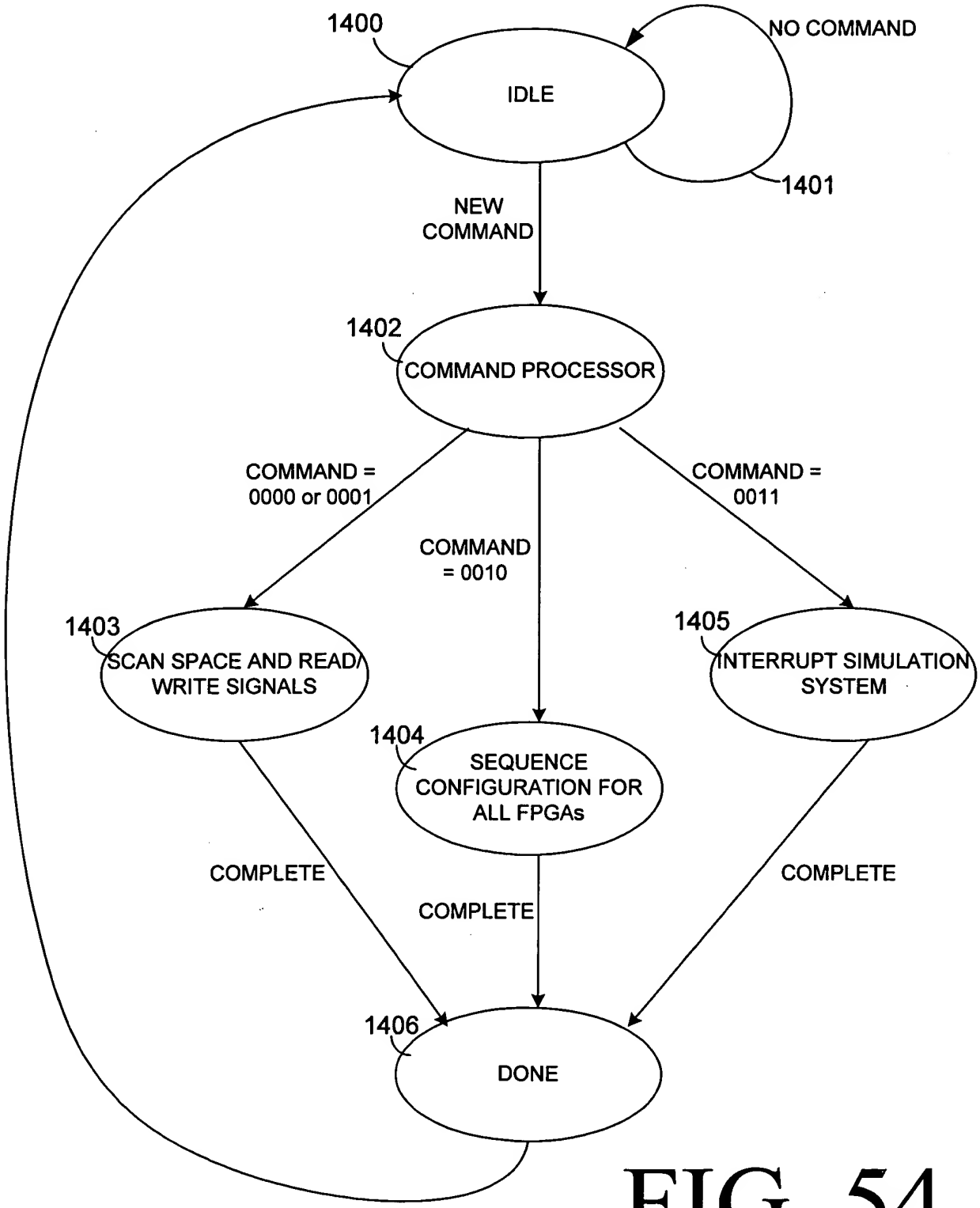


FIG 54

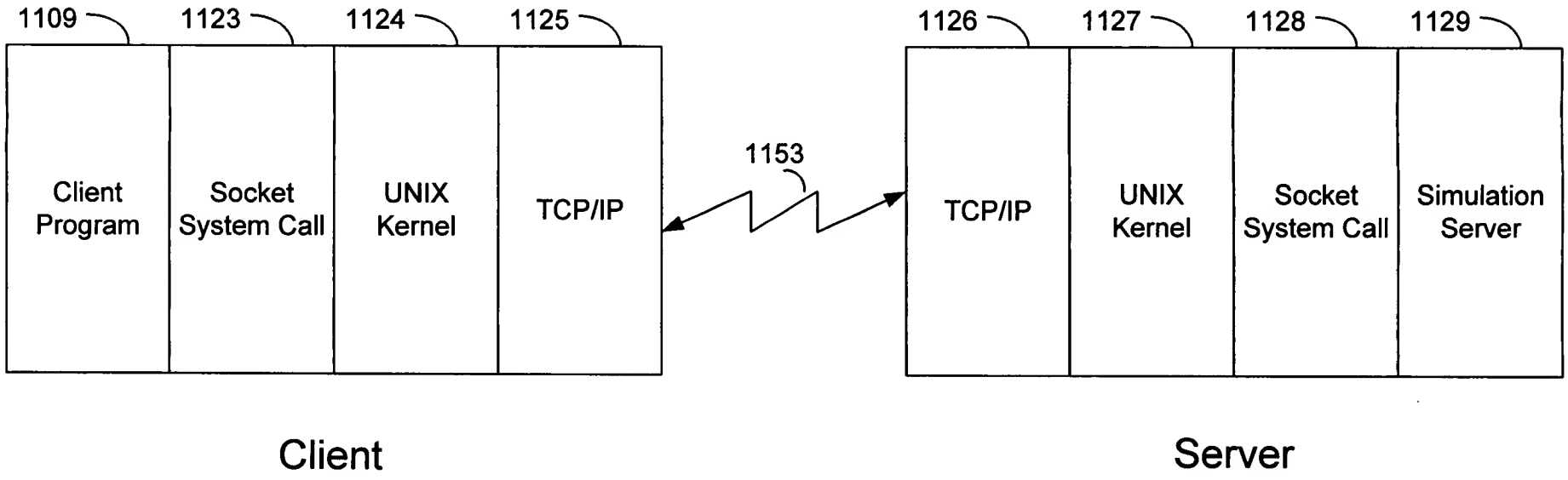

1109
1123
1124
1125
Client Program
Socket System Call
UNIX Kernel
TCP/IP
1153
1126
1127
1128
1129
TCP/IP
UNIX Kernel
Socket System Call
Simulation Server
Client
Server

FIG. 55

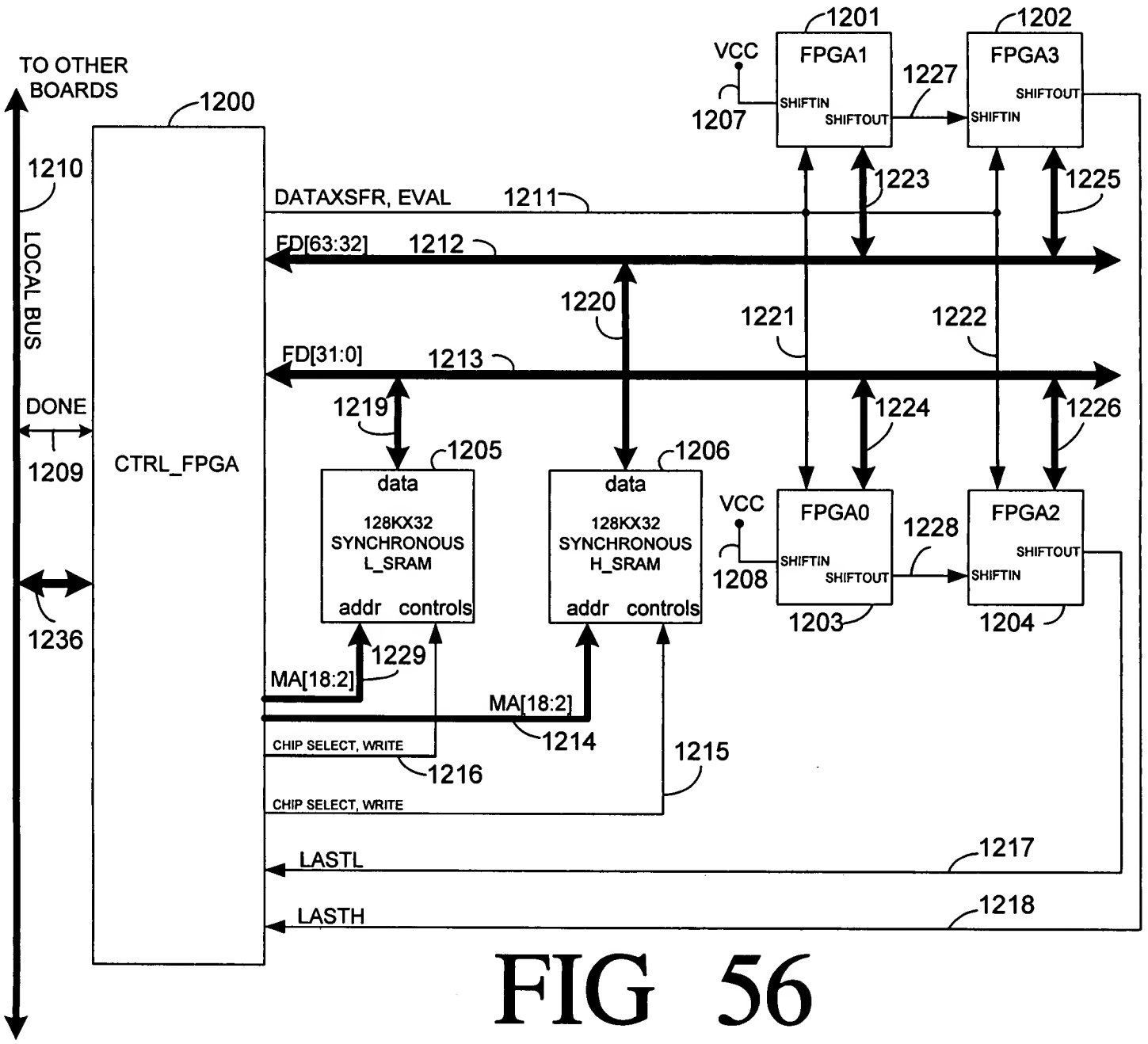

TO OTHER BOARDS
1200
1201
1202
VCC
FPGA1
FPGA3
1227
SHIFTIN
SHIFTOUT
1207
1210
LOCAL BUS
DATAXSFR, EVAL
1211
1223
1225
FD[63:32]
1212
1220
1221
1222
FD[31:0]
1213
DONE
1209
1219
1224
1226
CTRL_FPGA
1205
1206
data
128KX32
SYNCHRONOUS
L_SRAM
addr controls
data
128KX32
SYNCHRONOUS
H_SRAM
addr controls
VCC
FPGA0
FPGA2
1228
1208
1203
1204
1236
1229
MA[18:2]
MA[18:2]
1214
CHIP SELECT, WRITE
1216
1215
CHIP SELECT, WRITE
LASTL
1217
LASTH
1218

FIG 56

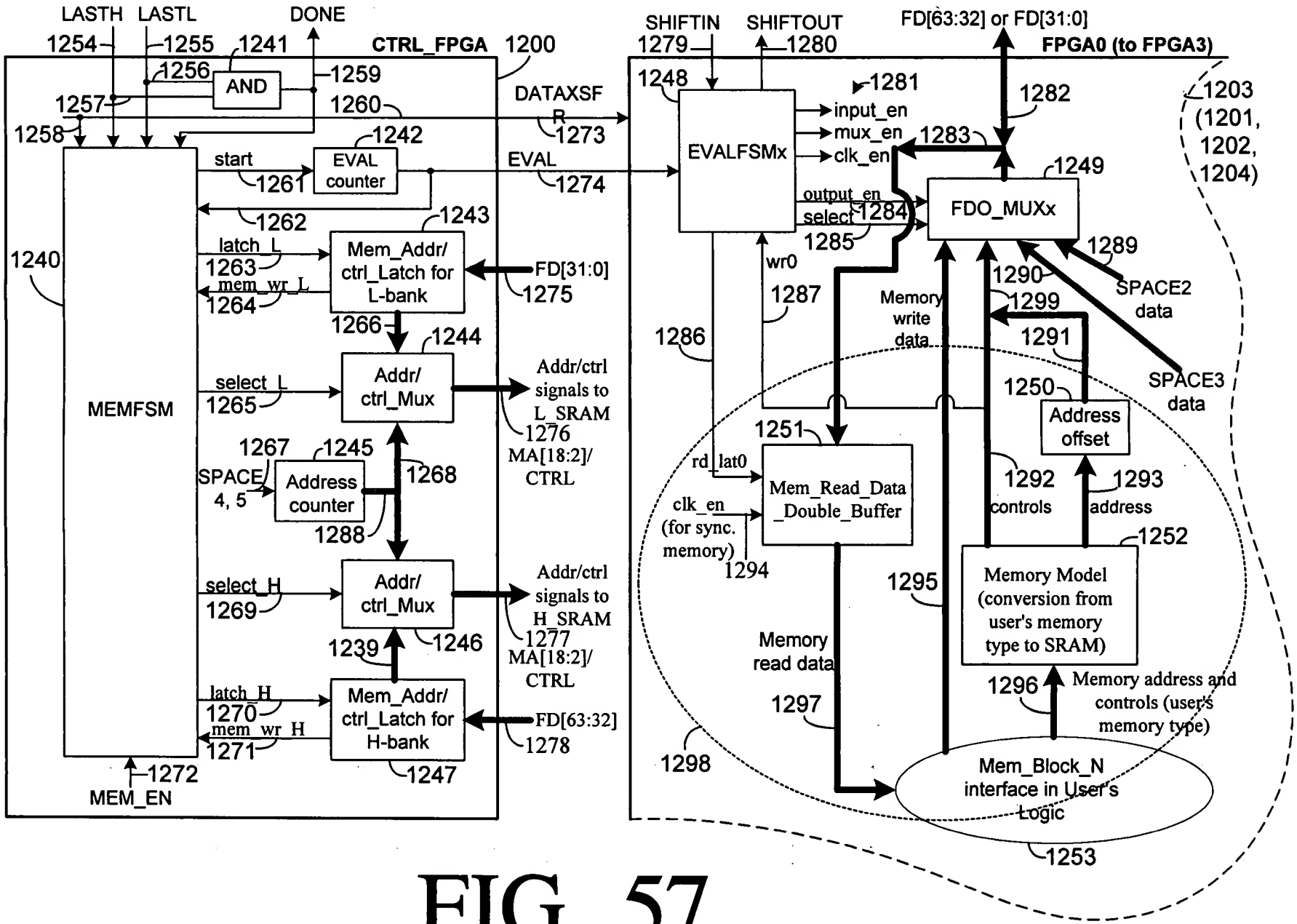

LASTH
LASTL
DONE
1254
1255
1241
CTRL_FPGA
1200
1256
AND
1259
1257
1260
1258
DATAXSF
R
1273
1242
start
EVAL
counter
EVAL
1274
1261
1262
1243
latch_L
1263
Mem_Addr/
ctrl_Latch for
L-bank
FD[31:0]
1275
mem_wr_L
1264
1240
1266
1244
select_L
1265
Addr/
ctrl_Mux
Addr/ctrl
signals to
L_SRAM
1276
MA[18:2]/
CTRL
MEMFSM
1267
1245
SPACE
4, 5
Address
counter
1268
1288
select_H
1269
Addr/
ctrl_Mux
Addr/ctrl
signals to
H_SRAM
1277
MA[18:2]/
CTRL
1239
1246
latch_H
1270
Mem_Addr/
ctrl_Latch for
H-bank
FD[63:32]
1278
mem_wr_H
1271
1247
1272
MEM_EN
SHIFTIN
1279
SHIFTOUT
1280
FD[63:32] or FD[31:0]
FPGA0 (to FPGA3)
1203
(1201,
1202,
1204)
1248
1281
EVALFSMx
input_en
mux_en
clk_en
1282
1283
1249
output_en
select
1284
1285
FDO_MUXx
wr0
1287
1289
1290
SPACE2
data
Memory
write
data
1299
1291
1286
SPACE3
data
1250
Address
offset
1251
rd_lat0
Mem_Read_Data
_Double_Buffer
1292
controls
1293
address
clk_en
(for sync.
memory)
1294
1252
1295
Memory Model
(conversion from
user's memory
type to SRAM)
Memory
read data
1296
Memory address and
controls (user's
memory type)
1297
1298
Mem_Block_N
interface in User's
Logic
1253

FIG 57

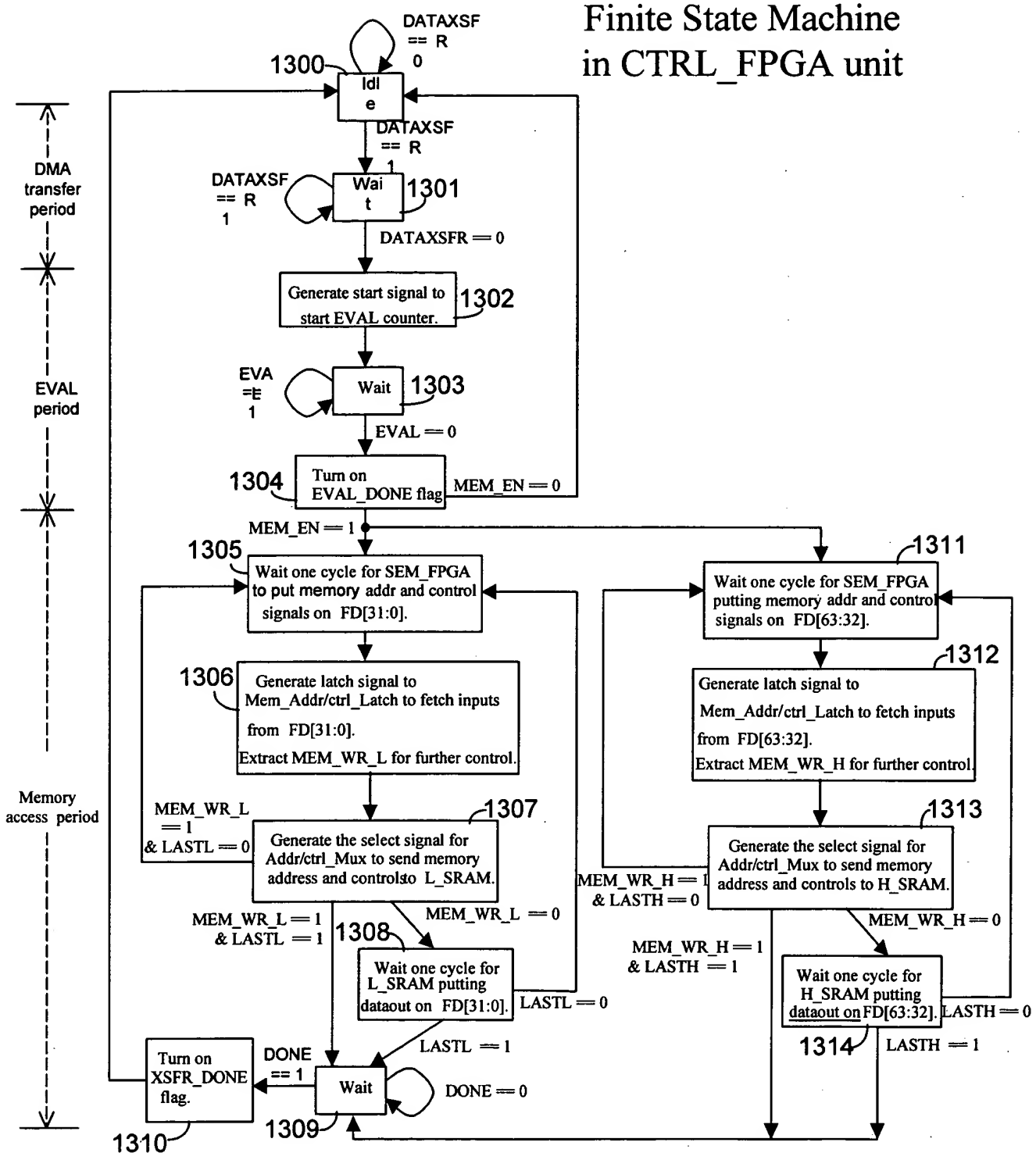

MEMFSM - Memory Finite State Machine in CTRL_FPGA unit
DMA transfer period
EVAL period
Memory access period
1300
Idle
DATAXSF == R 0
DATAXSF == R 1
1301
Wait
DATAXSF == R 1
DATAXSFR = 0
1302
Generate start signal to start EVAL counter.
1303
Wait
EVA =E 1
EVAL = 0
1304
Turn on EVAL_DONE flag
MEM_EN = 0
MEM_EN = 1
1305
Wait one cycle for SEM_FPGA to put memory addr and control signals on FD[31:0].
1306
Generate latch signal to Mem_Addr/ctrl_Latch to fetch inputs from FD[31:0].
Extract MEM_WR_L for further control.
1307
MEM_WR_L = 1 & LASTL = 0
Generate the select signal for Addr/ctrl_Mux to send memory address and controlsto L_SRAM.
MEM_WR_L = 1 & LASTL = 1
MEM_WR_L = 0
1308
Wait one cycle for L_SRAM putting dataout on FD[31:0].
LASTL = 0
LASTL = 1
DONE == 1
Turn on XSFR_DONE flag.
Wait
DONE = 0
1309
1310
1311
Wait one cycle for SEM_FPGA putting memory addr and control signals on FD[63:32].
1312
Generate latch signal to Mem_Addr/ctrl_Latch to fetch inputs from FD[63:32].
Extract MEM_WR_H for further control.
1313
Generate the select signal for Addr/ctrl_Mux to send memory address and controls to H_SRAM.
MEM_WR_H = 1 & LASTH = 0
MEM_WR_H = 0
MEM_WR_H = 1 & LASTH = 1
Wait one cycle for H_SRAM putting dataout on FD[63:32].
LASTH = 0
LASTH = 1
1314

FIG 58

EVALFSM - EVAL
Finite State Machine in each FPGA logic device

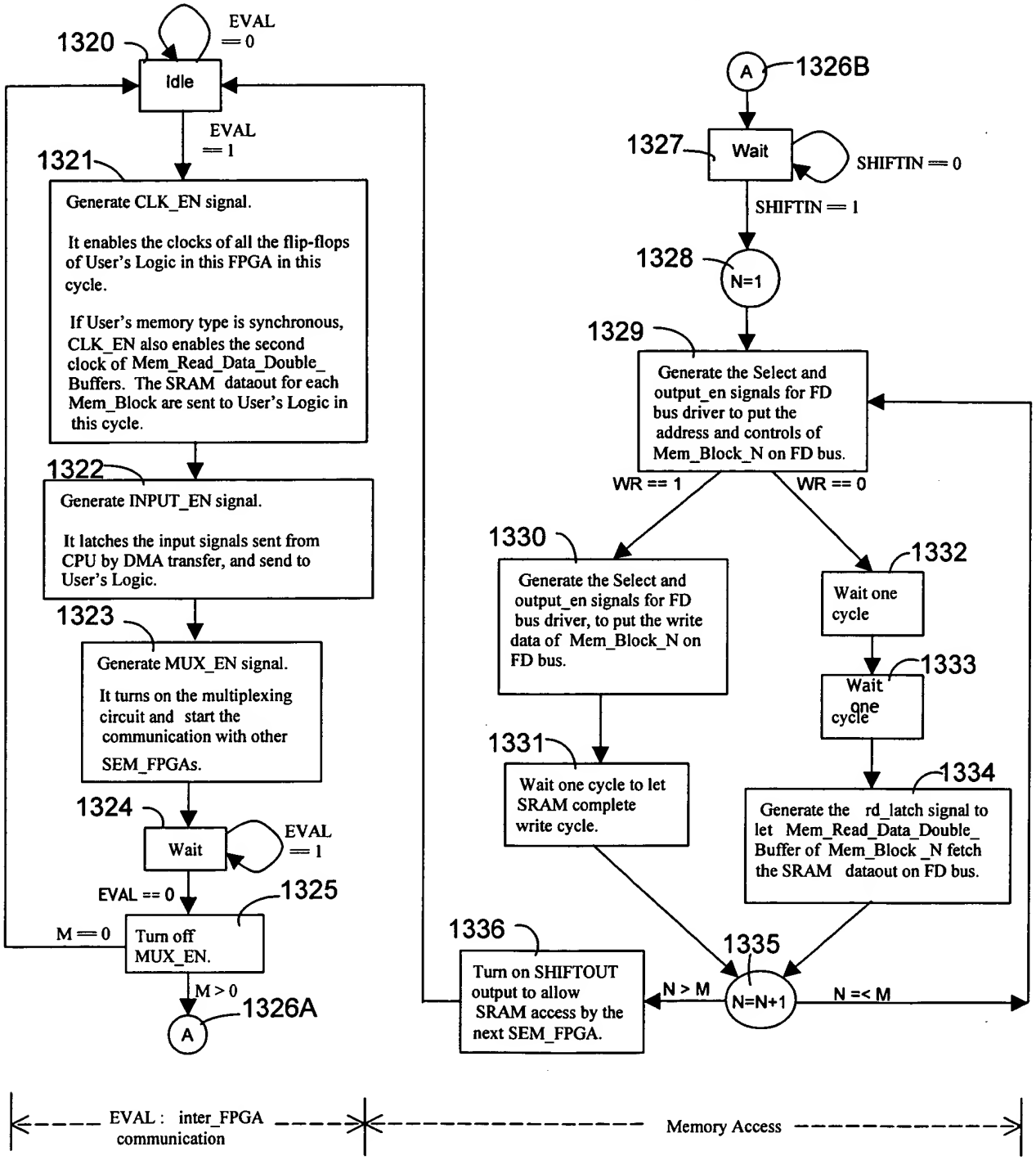


FIG 59

MEMORY READ DATA DOUBLE BUFFER

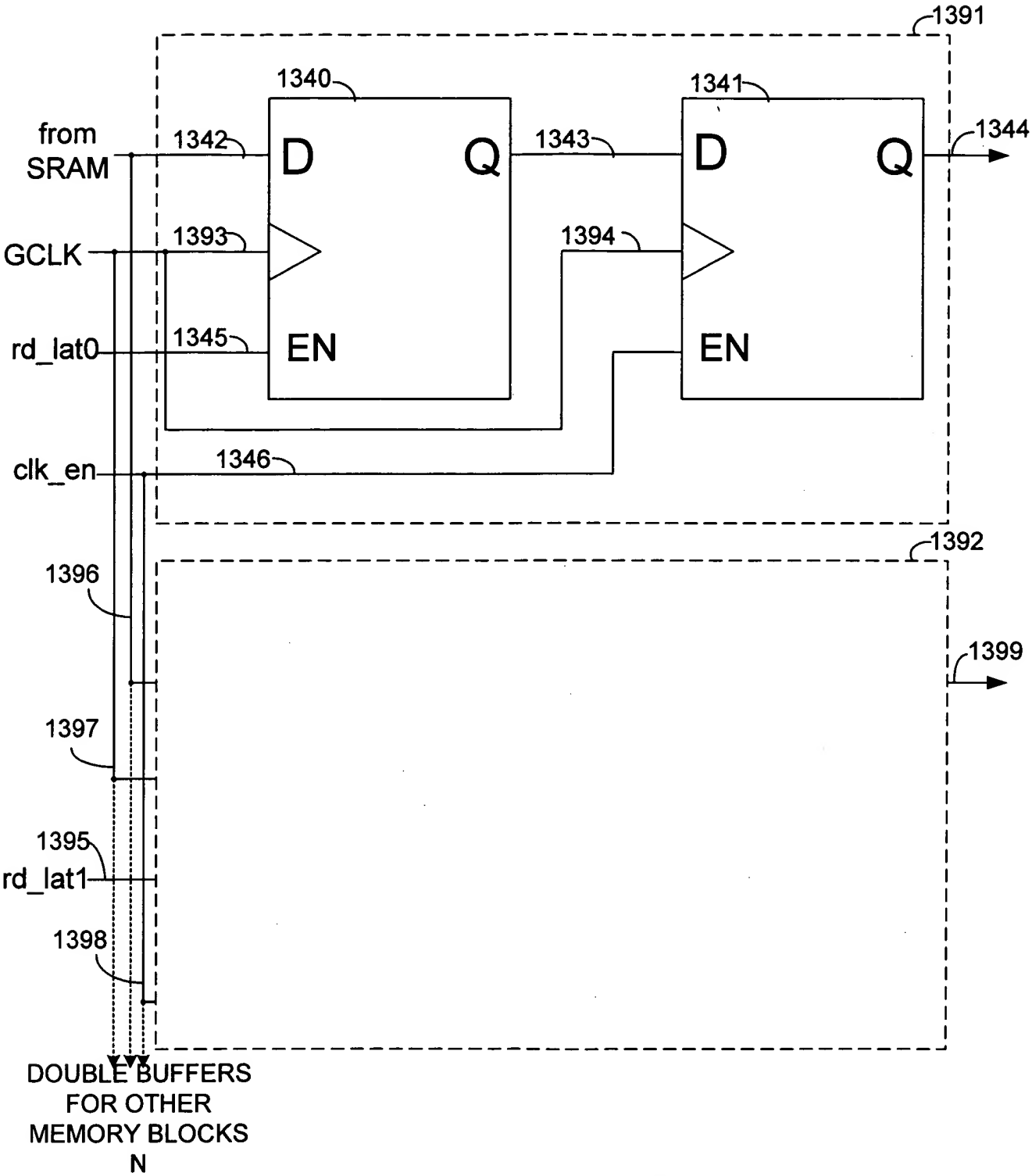


FIG 60

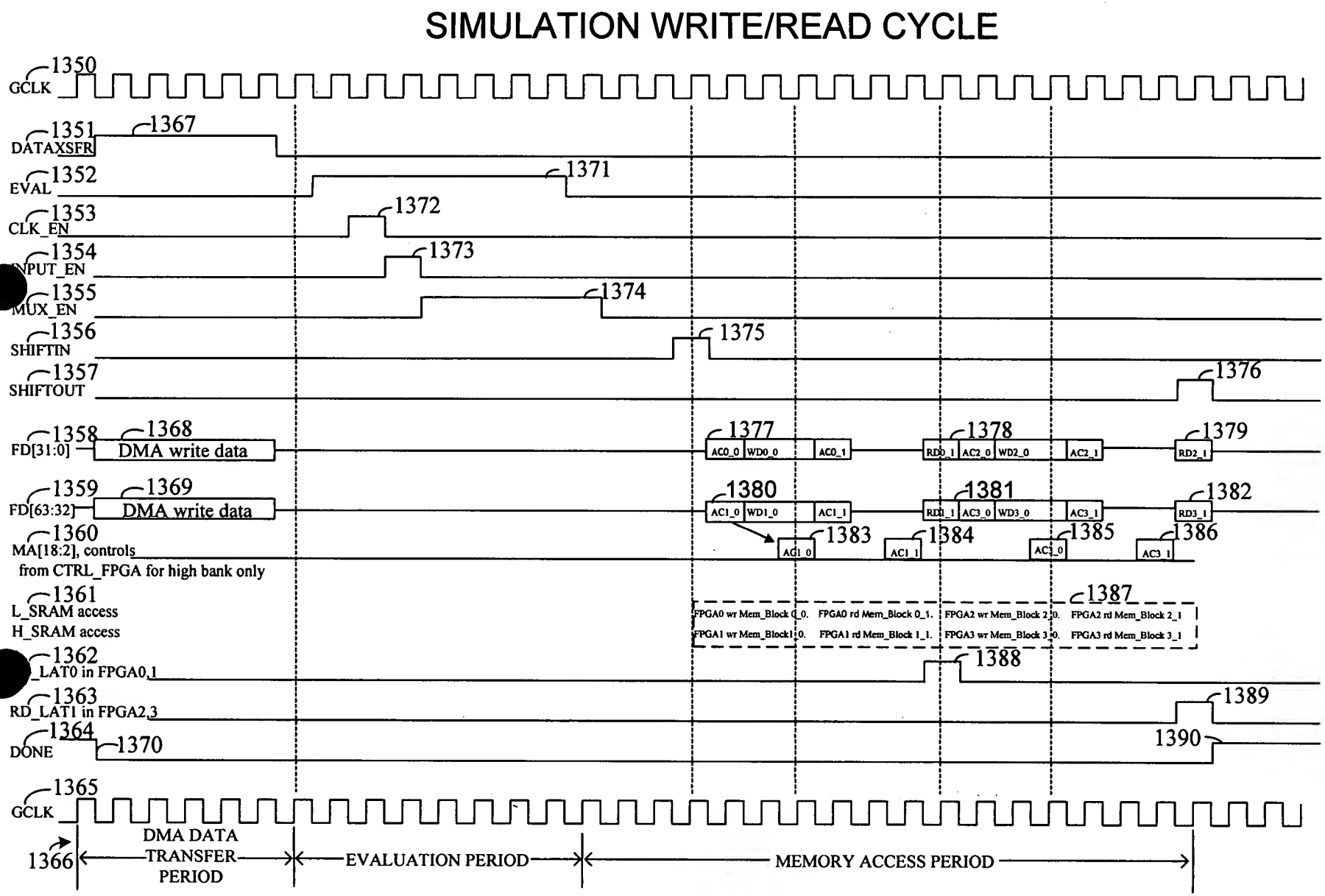


FIG 61

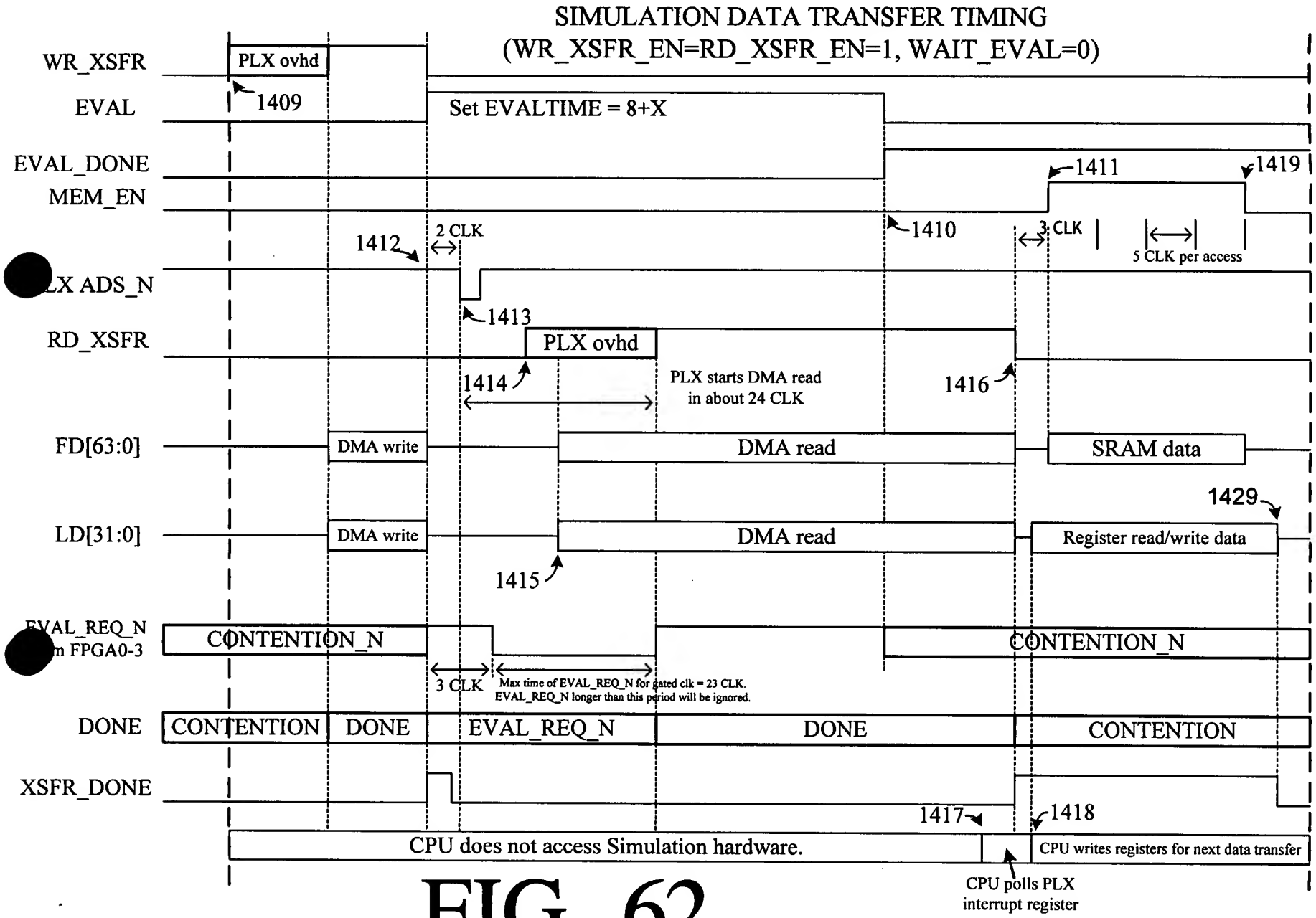

SIMULATION DATA TRANSFER TIMING
(WR_XSFR_EN=RD_XSFR_EN=1, WAIT_EVAL=0)
WR_XSFR
PLX ovhd
EVAL
1409
Set EVALTIME = 8+X
EVAL_DONE
1411
1419
MEM_EN
1410
3 CLK
5 CLK per access
1412
2 CLK
LX ADS_N
1413
RD_XSFR
PLX ovhd
1414
PLX starts DMA read
in about 24 CLK
1416
FD[63:0]
DMA write
DMA read
SRAM data
1429
LD[31:0]
DMA write
DMA read
Register read/write data
1415
VAL_REQ_N
m FPGA0-3
CONTENTION_N
CONTENTION_N
3 CLK
Max time of EVAL_REQ_N for gated clk = 23 CLK.
EVAL_REQ_N longer than this period will be ignored.
DONE
CONTENTION
DONE
EVAL_REQ_N
DONE
CONTENTION
XSFR_DONE
1417
1418
CPU does not access Simulation hardware.
CPU writes registers for next data transfer
CPU polls PLX
interrupt register

FIG. 62

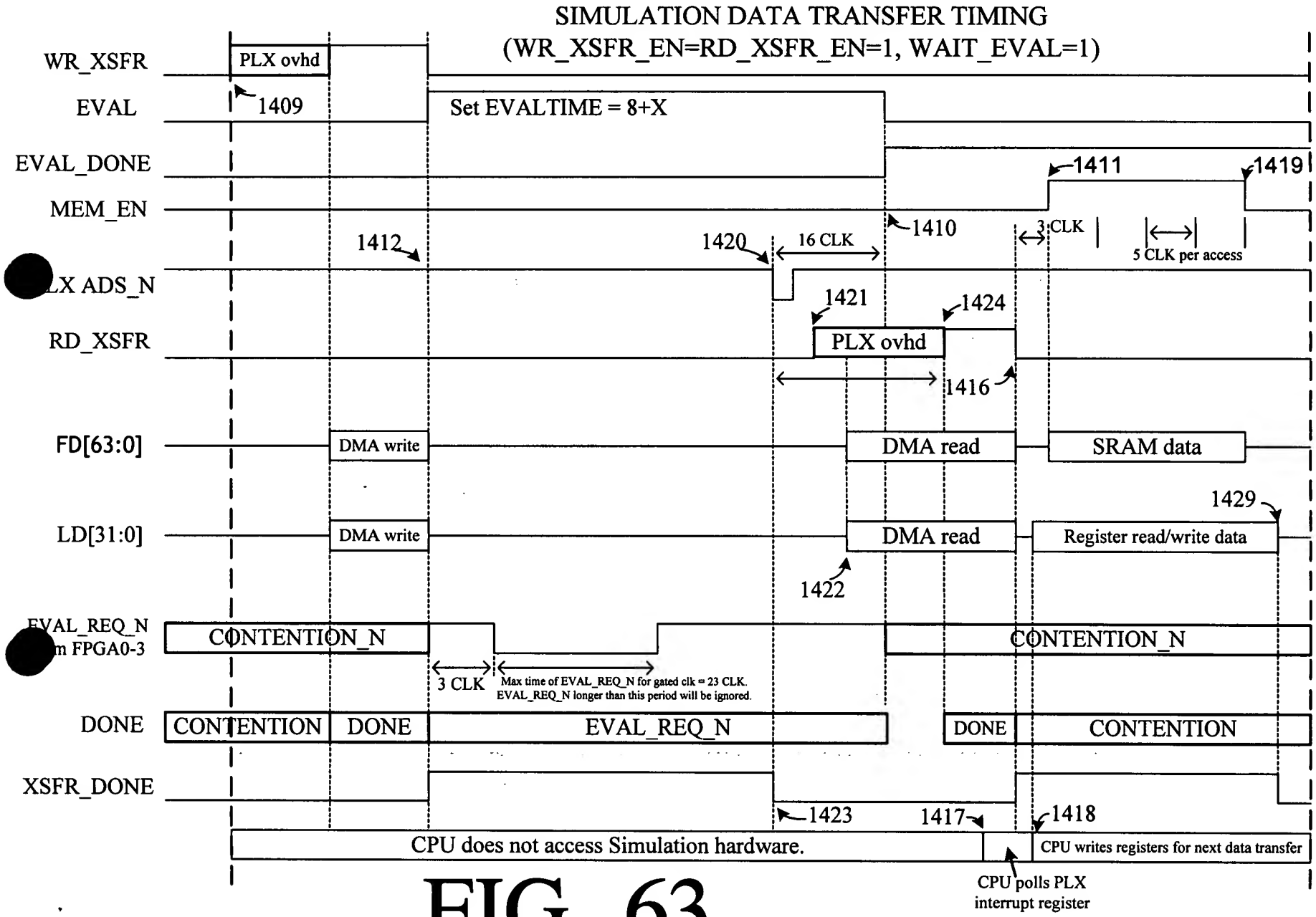


FIG. 63

Typical User Design of PCI Add-on Cards

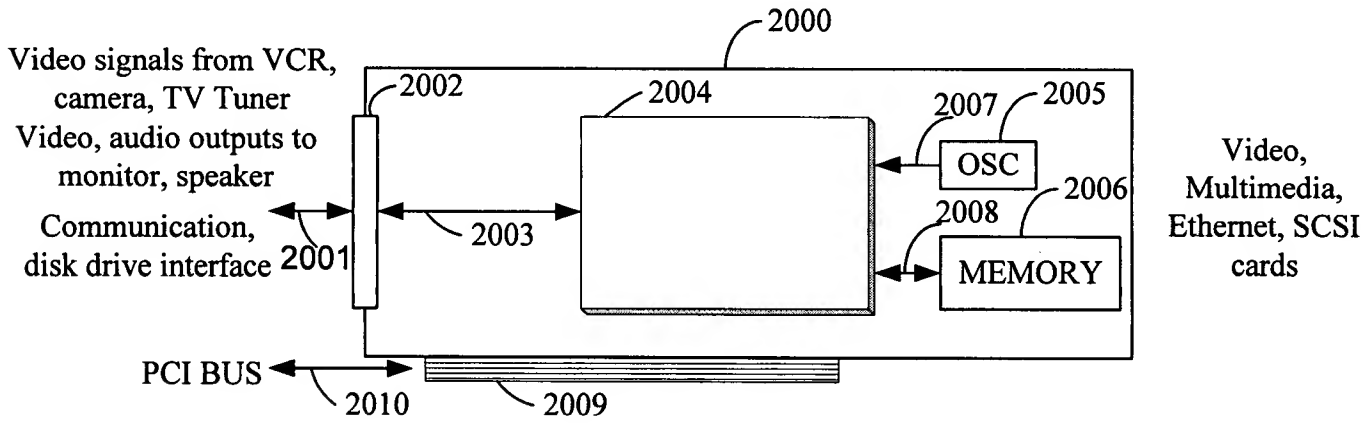


FIG. 64

Typical Hardware/Software Co-Verification

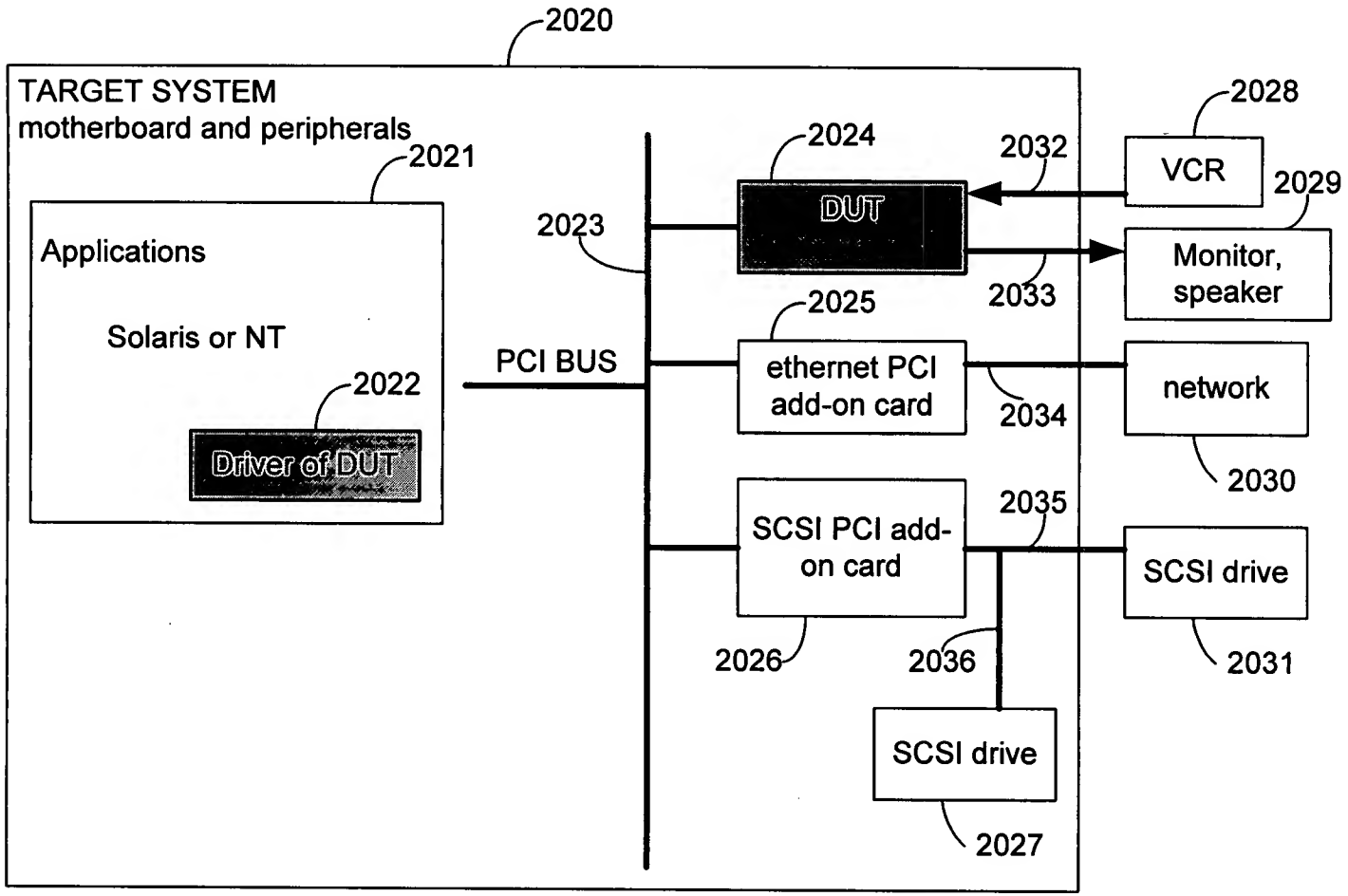
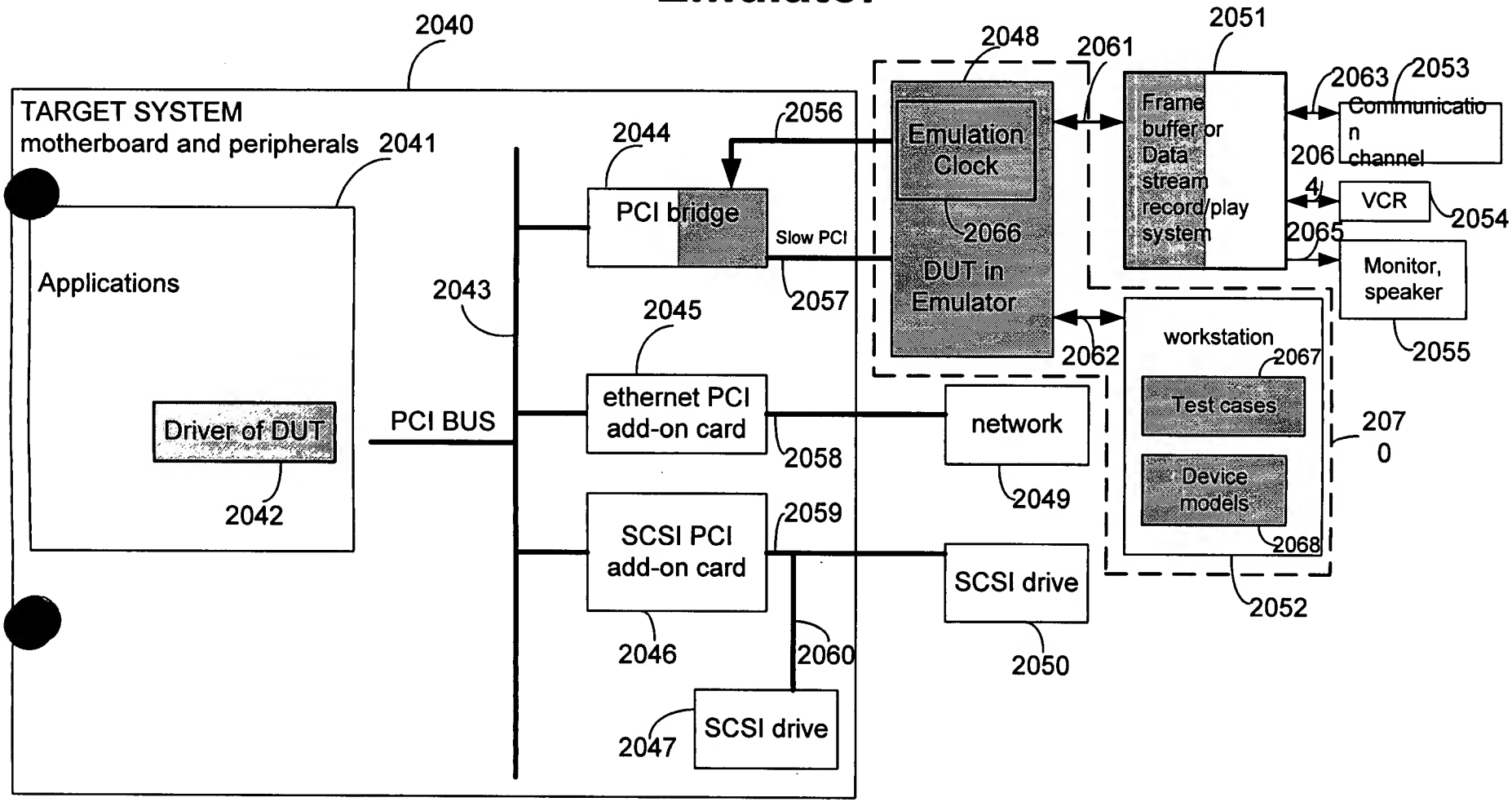


FIG. 65

Typical Co-Verification by Using Emulator



: running time at emulation speed

The rest of the target system is running at full speed.

FIG. 66

SIMULATION

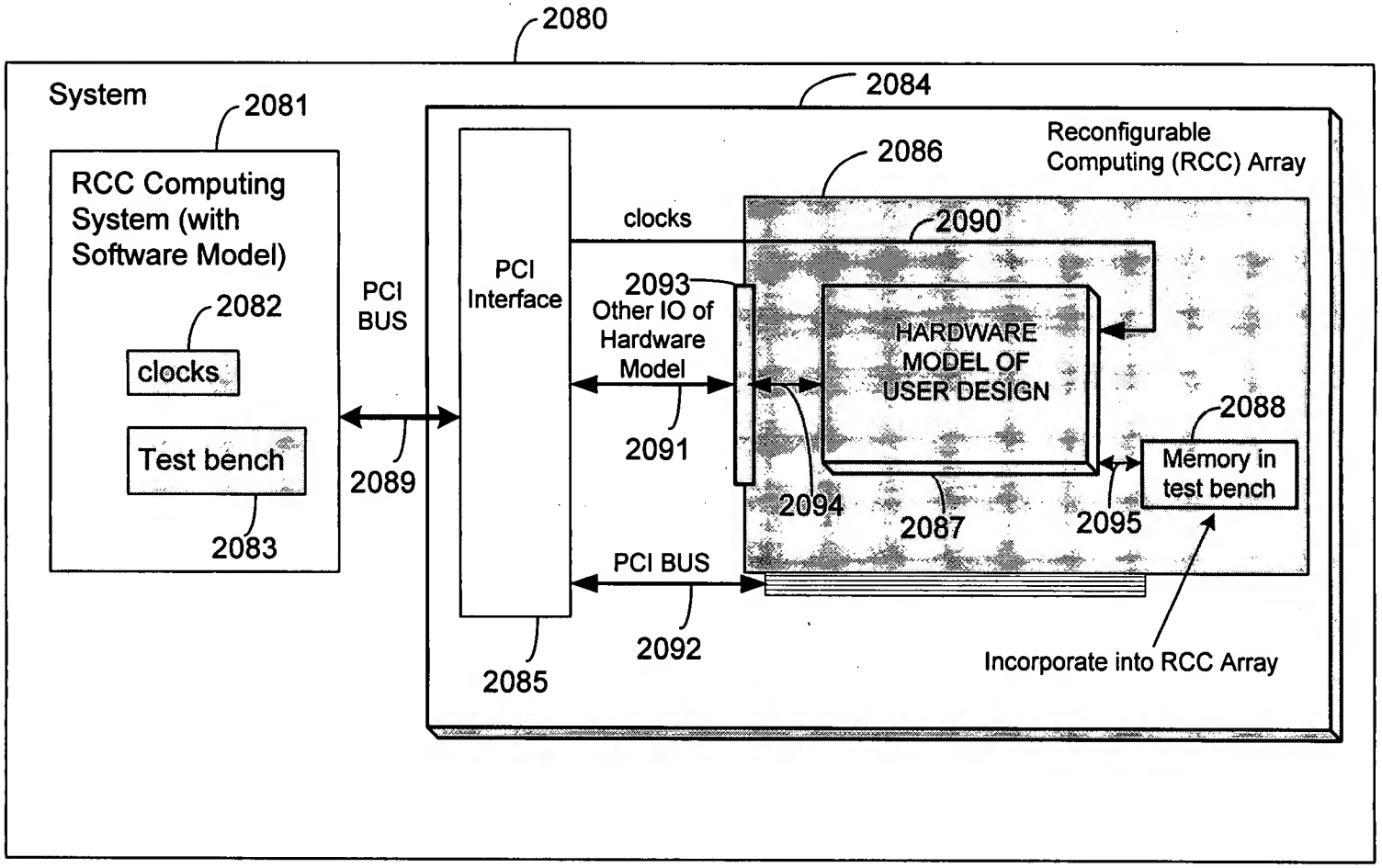


FIG. 67

CO-VERIFICATION WITHOUT EXTERNAL I/O

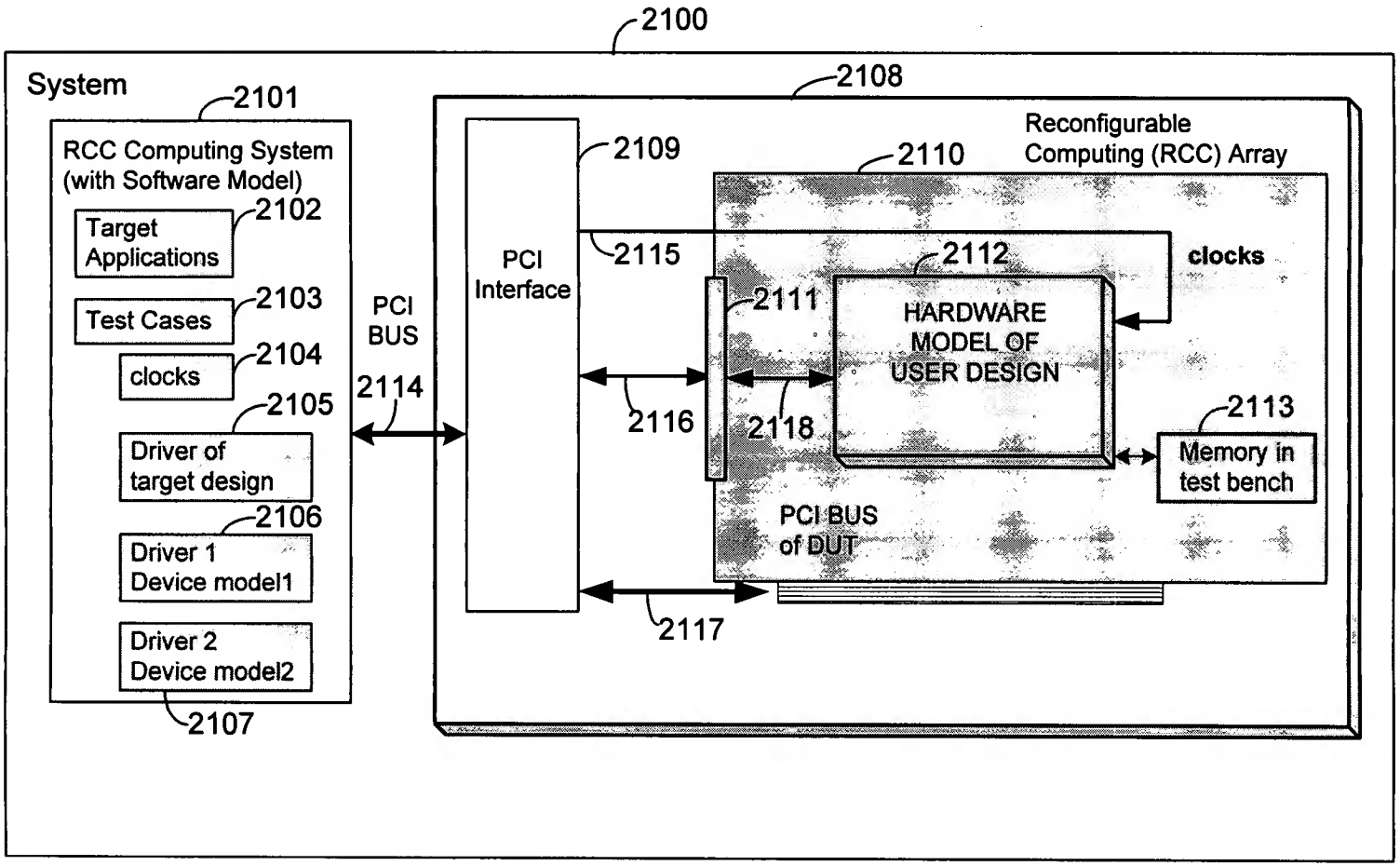


FIG. 68

CO-VERIFICATION WITH EXTERNAL I/O

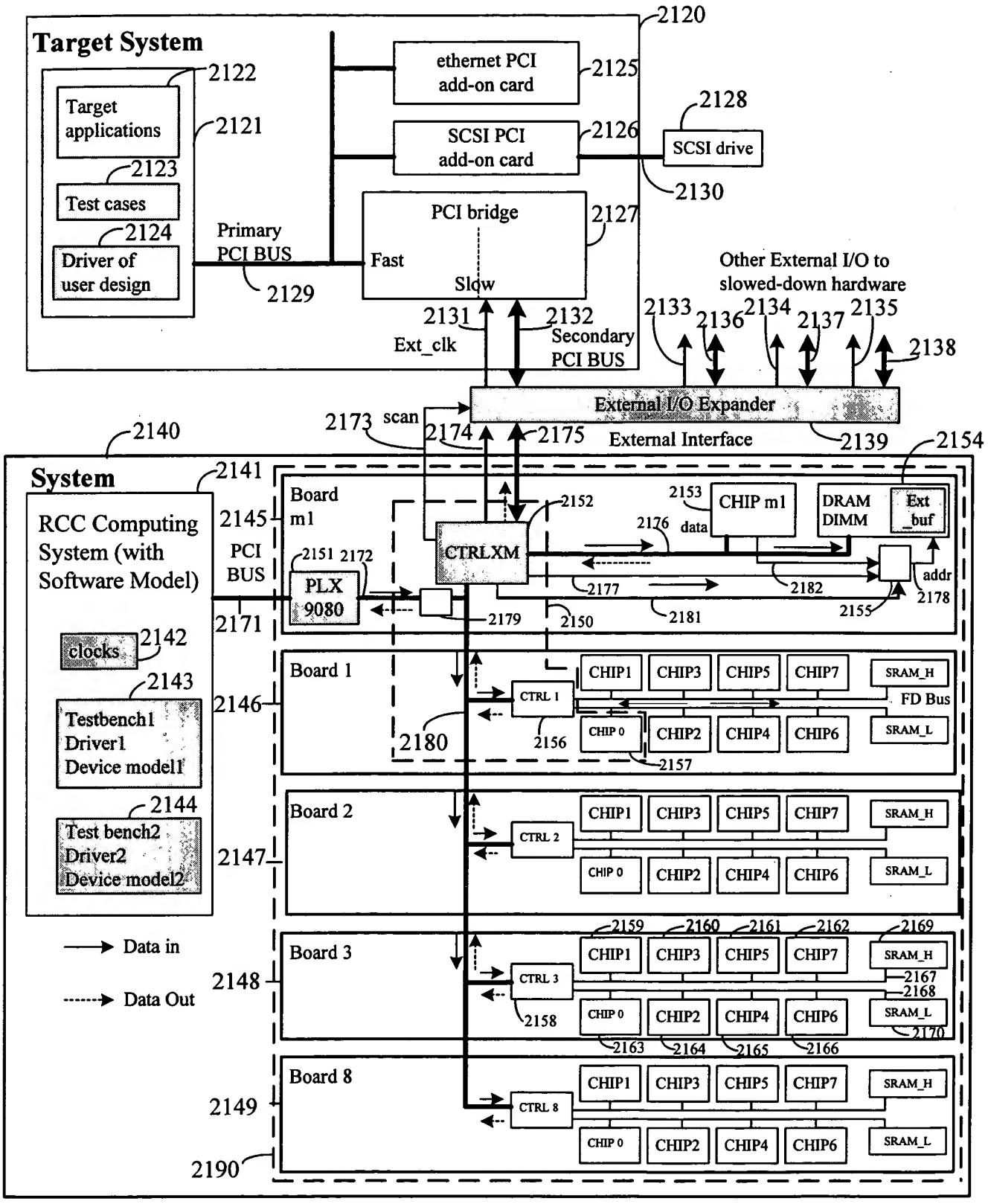


FIG. 69

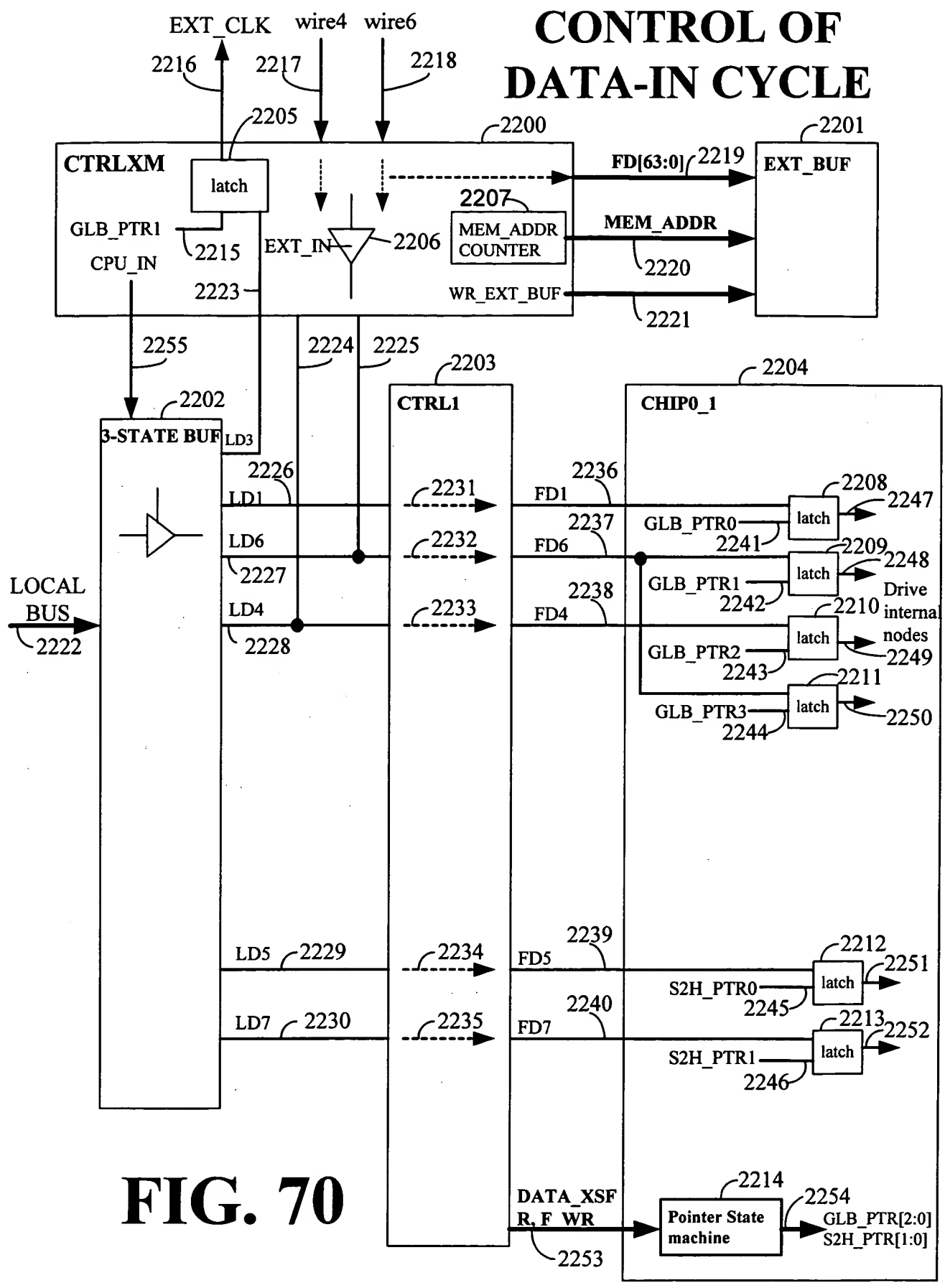

CONTROL OF DATA-IN CYCLE
EXT_CLK
wire4
wire6
2216
2217
2218
2205
2200
2201
CTRLXM
latch
FD[63:0]
2219
EXT_BUF
2207
GLB_PTR1
2215
CPU_IN
EXT_IN
2206
MEM_ADDR COUNTER
MEM_ADDR
2220
2223
WR_EXT_BUF
2221
2255
2224
2225
2203
2204
2202
3-STATE BUF
LD3
CTRL1
CHIP0_1
2226
2236
LD1
2231
FD1
2208
2247
GLB_PTR0
2241
2237
LD6
2232
FD6
2209
2248
GLB_PTR1
2242
LOCAL BUS
2227
2238
LD4
2233
FD4
2210
Drive internal nodes
2249
GLB_PTR2
2243
2211
GLB_PTR3
2244
2250
2222
2228
2239
LD5
2229
2234
FD5
2212
2251
S2H_PTR0
2245
2240
LD7
2230
2235
FD7
2213
2252
S2H_PTR1
2246
2214
DATA_XSF
R, F_WR
2253
Pointer State machine
2254
GLB_PTR[2:0]
S2H_PTR[1:0]

FIG. 70

CONTROL OF DATA-OUT CYCLE

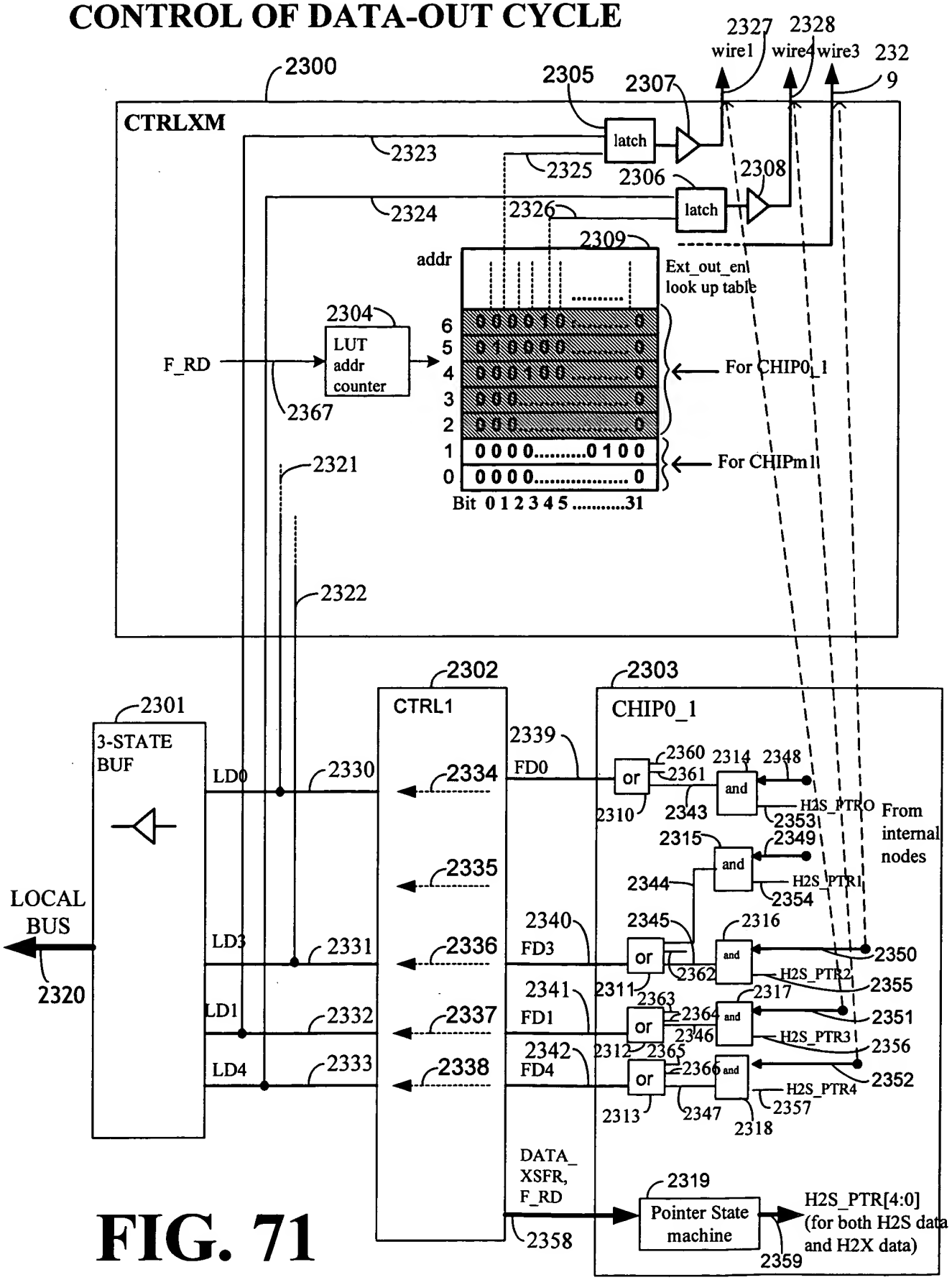


FIG. 71

CONTROL OF DATA-IN CYCLE

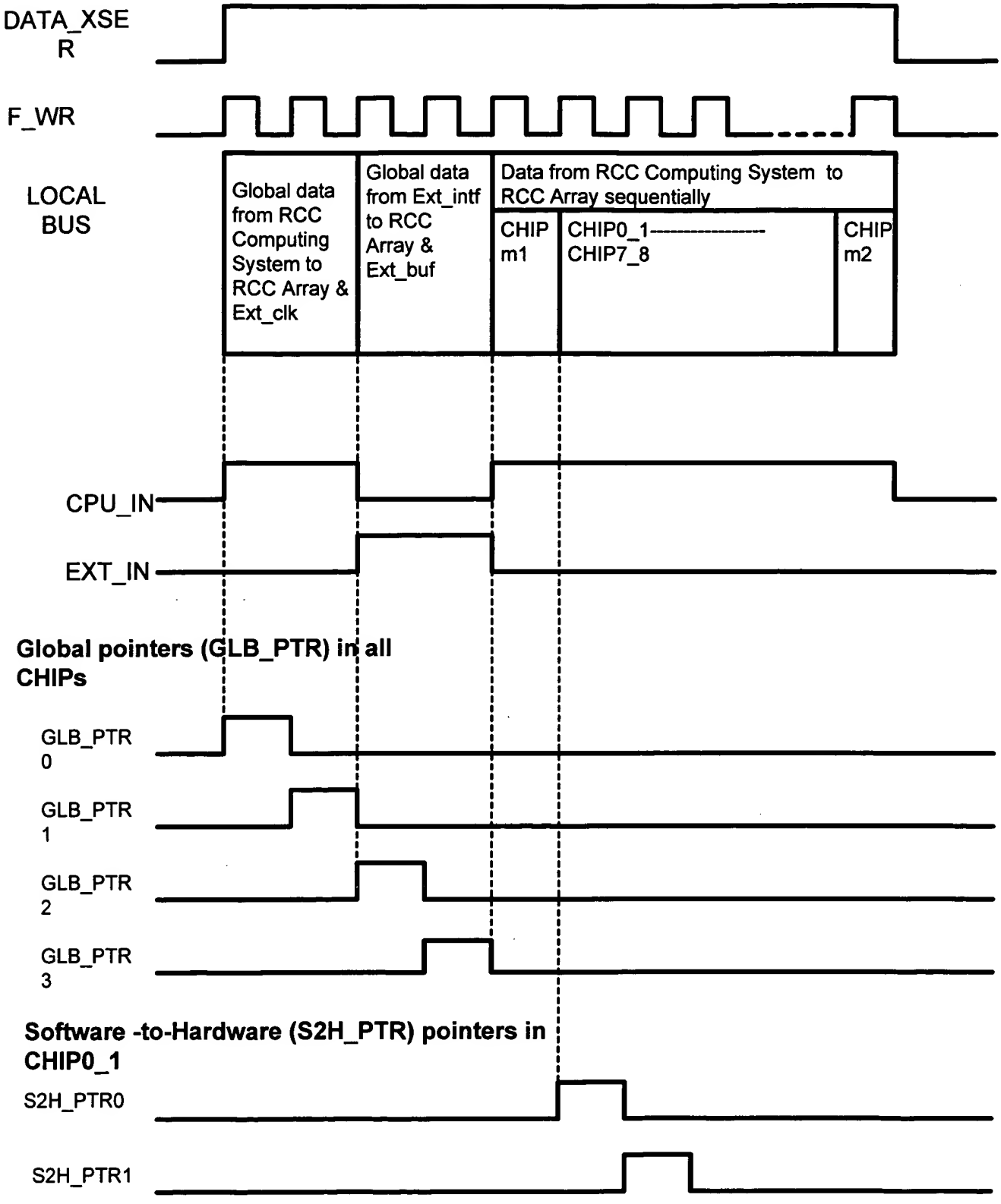


FIG. 72

CONTROL OF DATA-OUT CYCLE

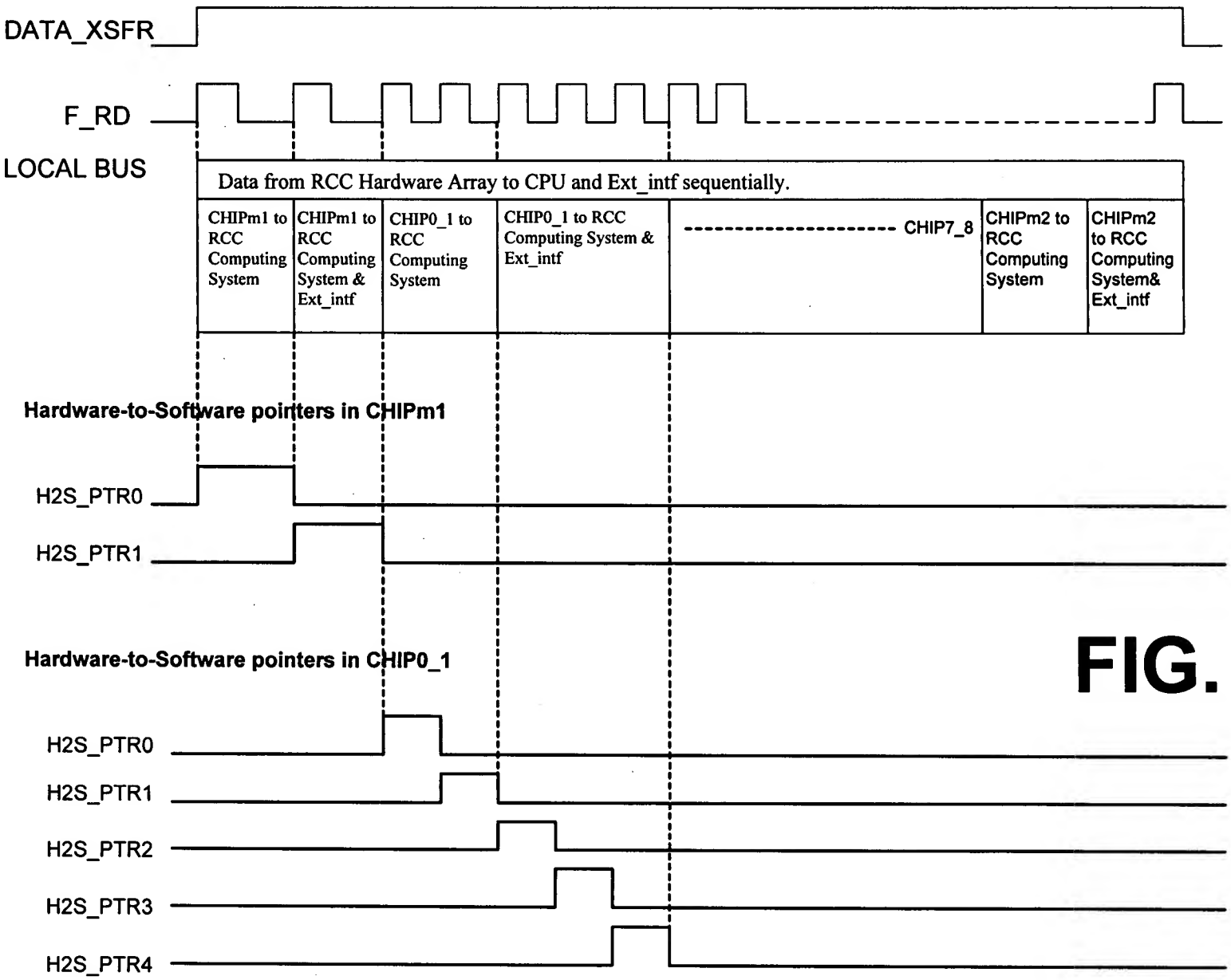


FIG. 73

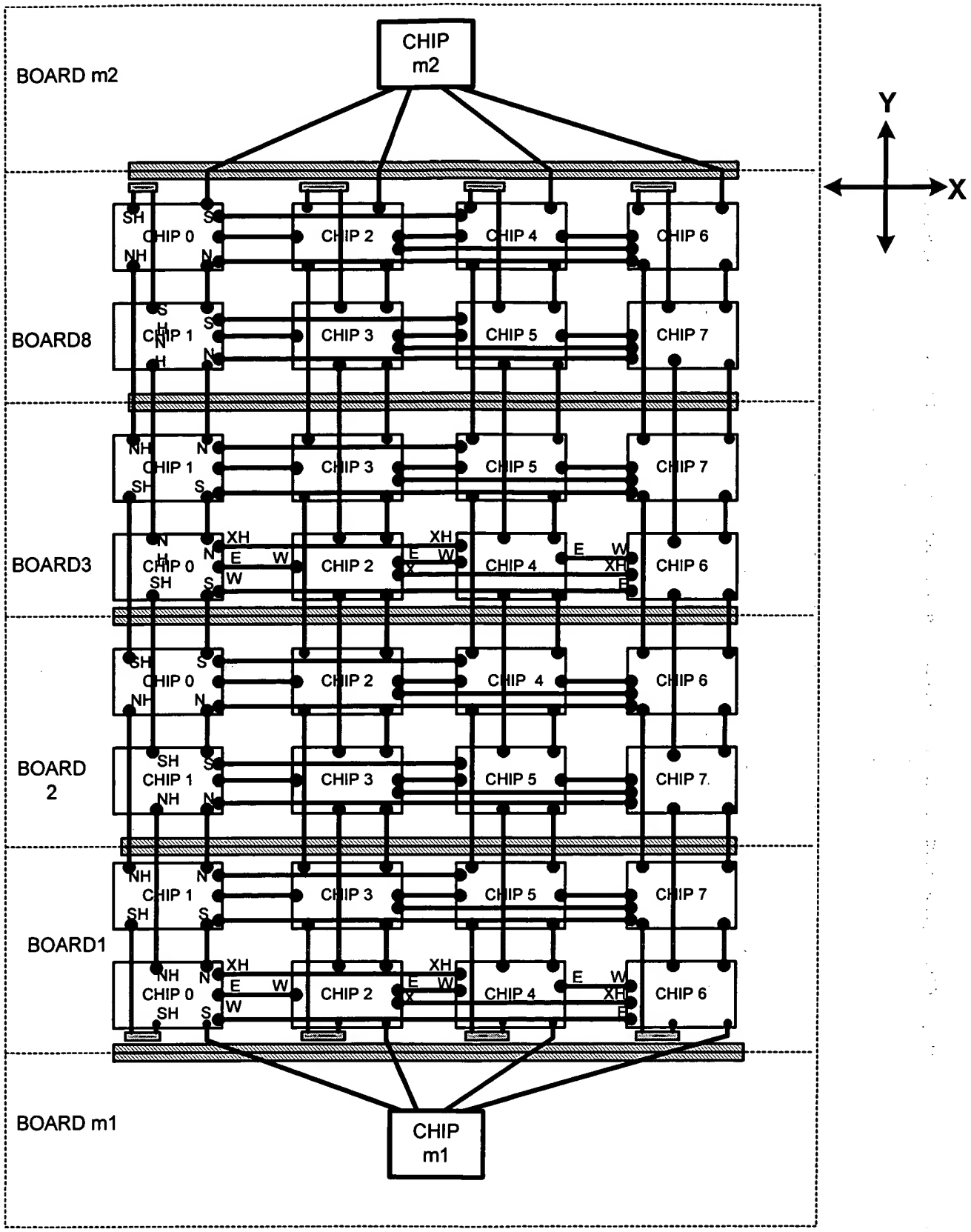

CHIP m2
BOARD m2
Y
X
CHIP 0
CHIP 2
CHIP 4
CHIP 6
BOARD8
CHIP 1
CHIP 3
CHIP 5
CHIP 7
BOARD3
BOARD 2
BOARD1
CHIP m1
BOARD m1

FIG. 74

SHIFT REGISTER

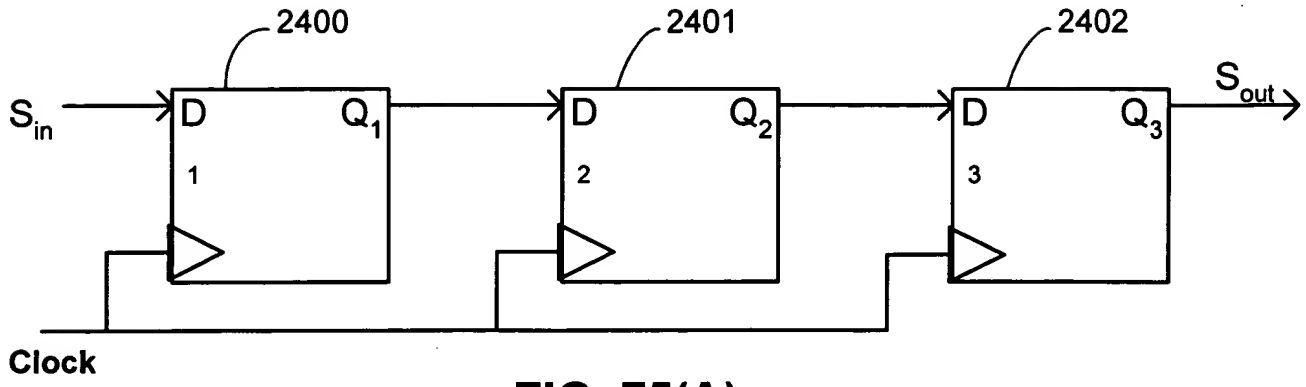


FIG. 75(A)

HOLD TIME ASSUMPTION FOR SHIFT REGISTER

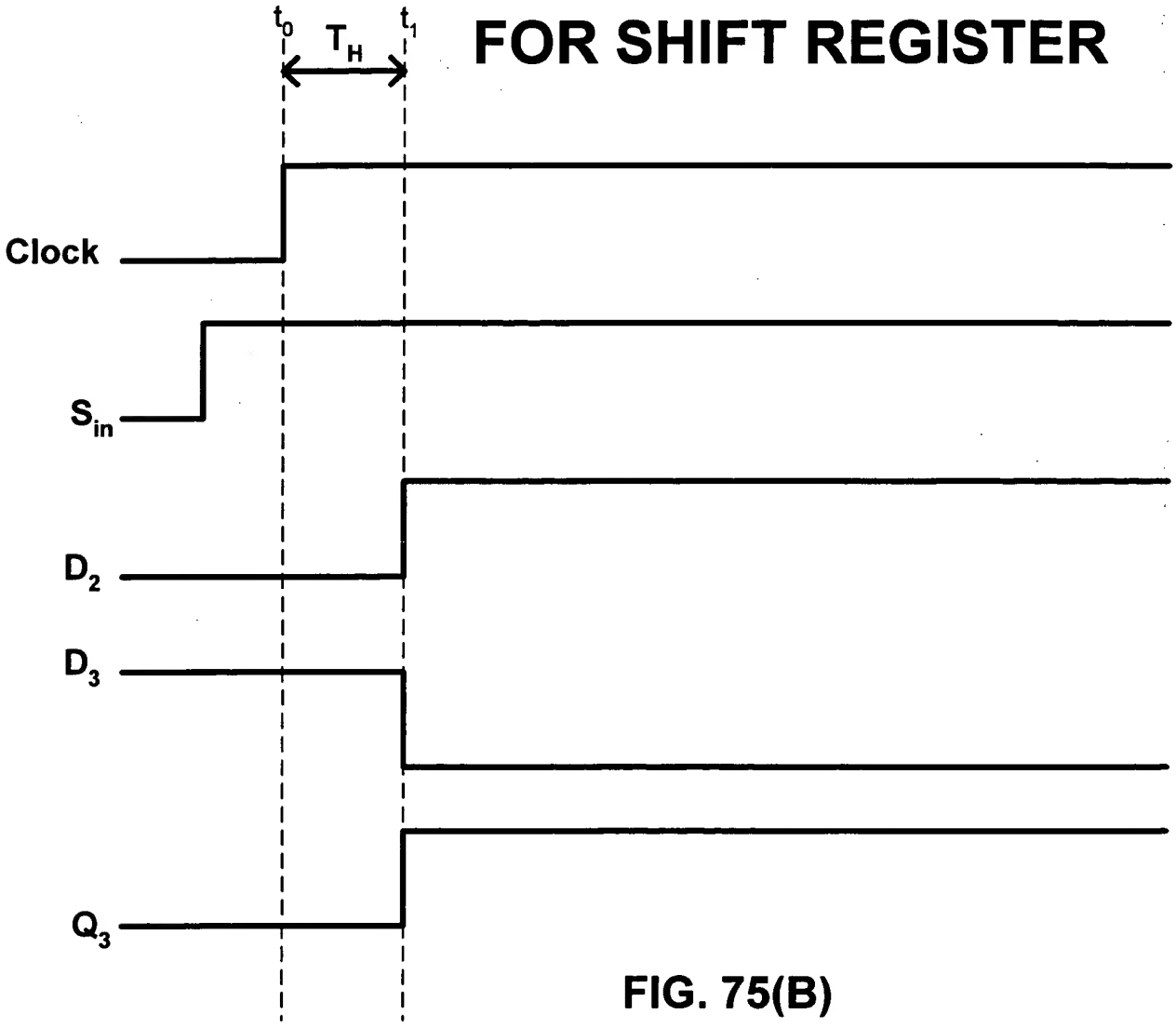


FIG. 75(B)

MULTIPLE FPGA MAPPING FOR SHIFT REGISTER

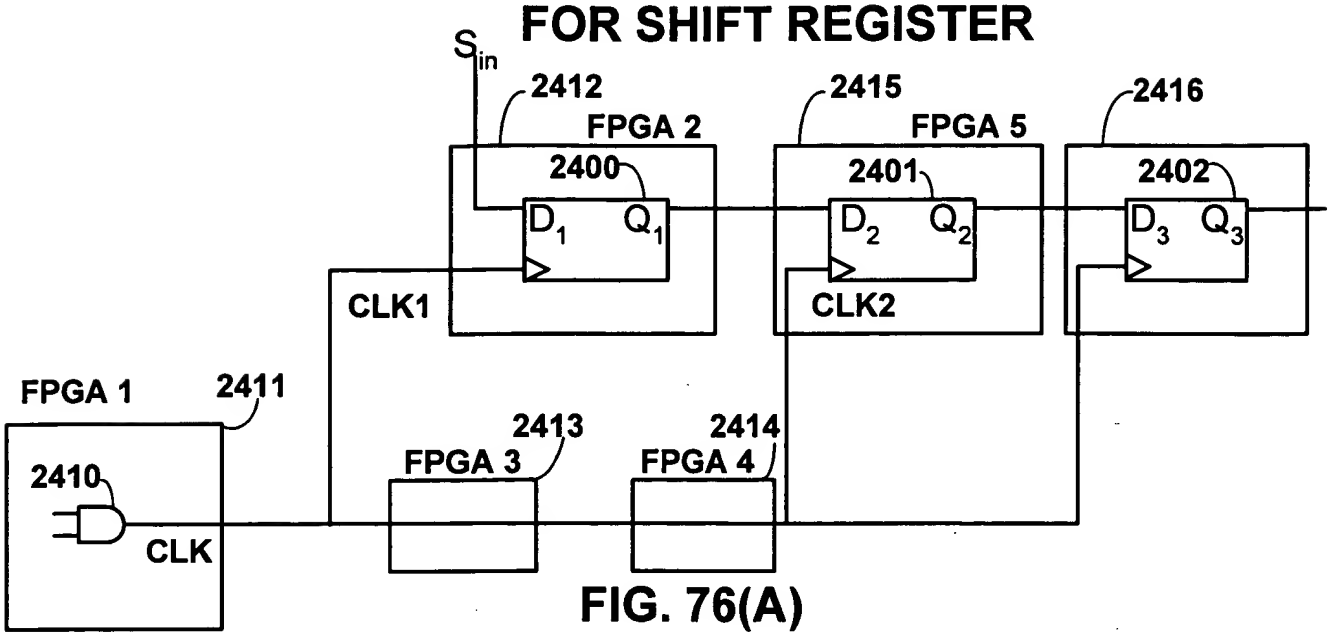
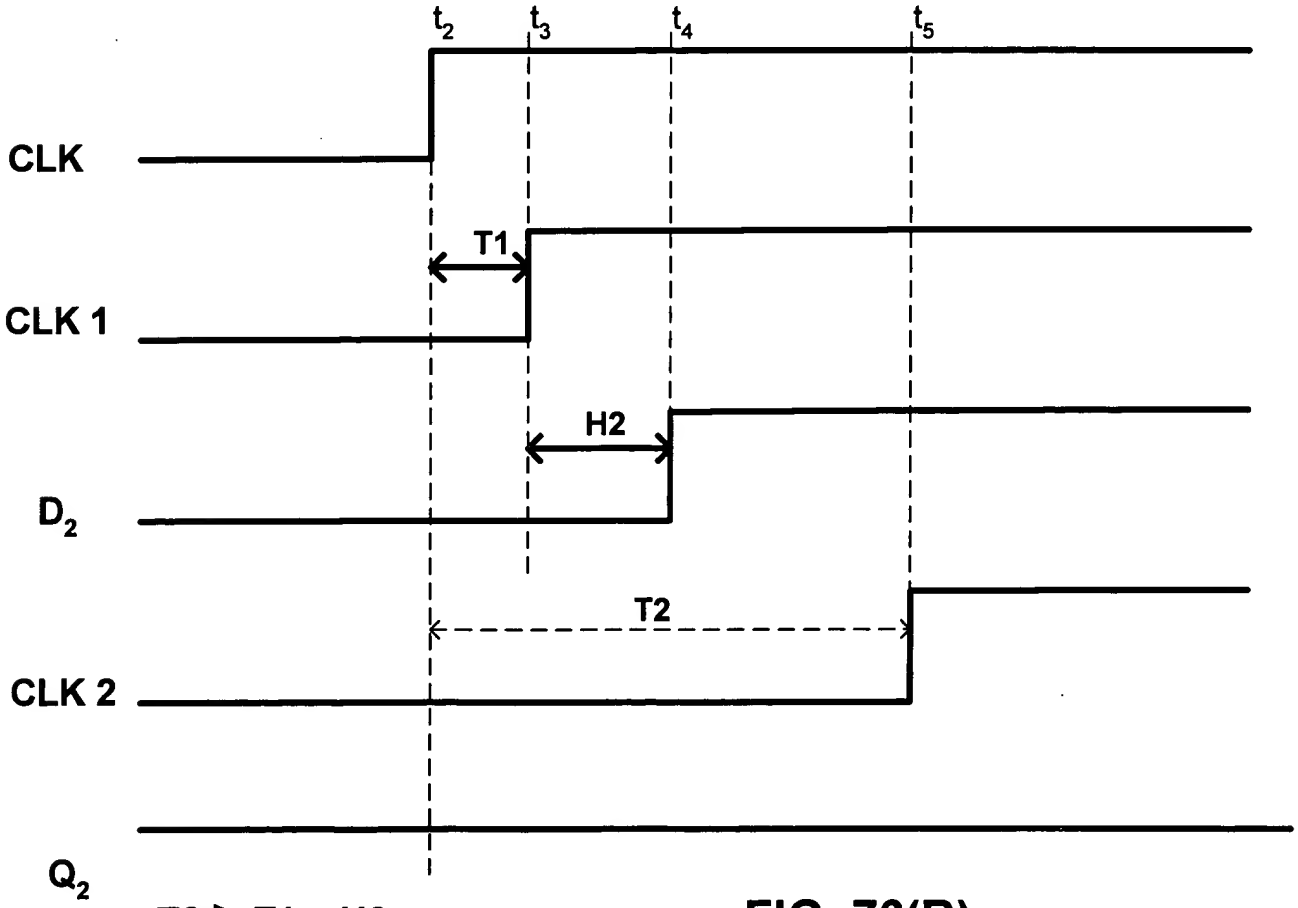


FIG. 76(A)

HOLD TIME VIOLATION BY LONG CLOCK SKEW



$T2 > T1 + H2$

FIG. 76(B)

CLOCK GLITCH PROBLEM

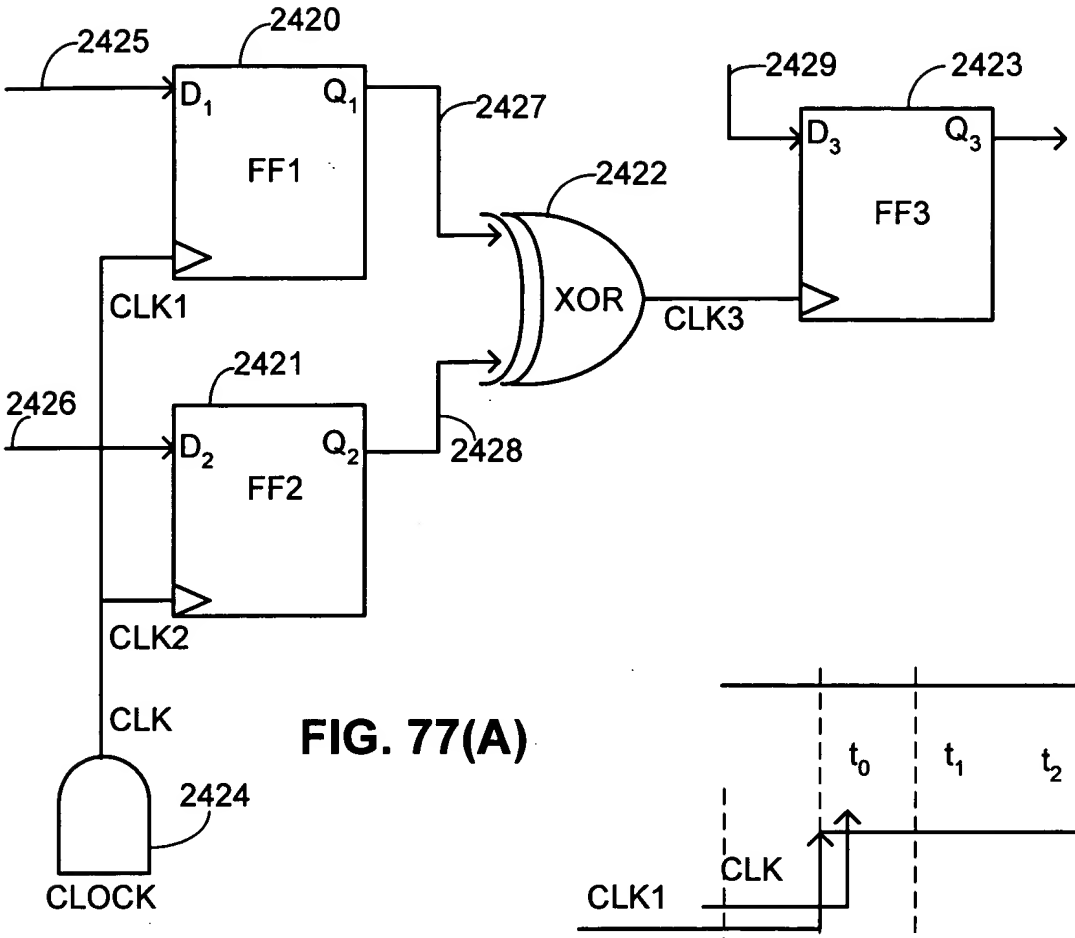


FIG. 77(A)

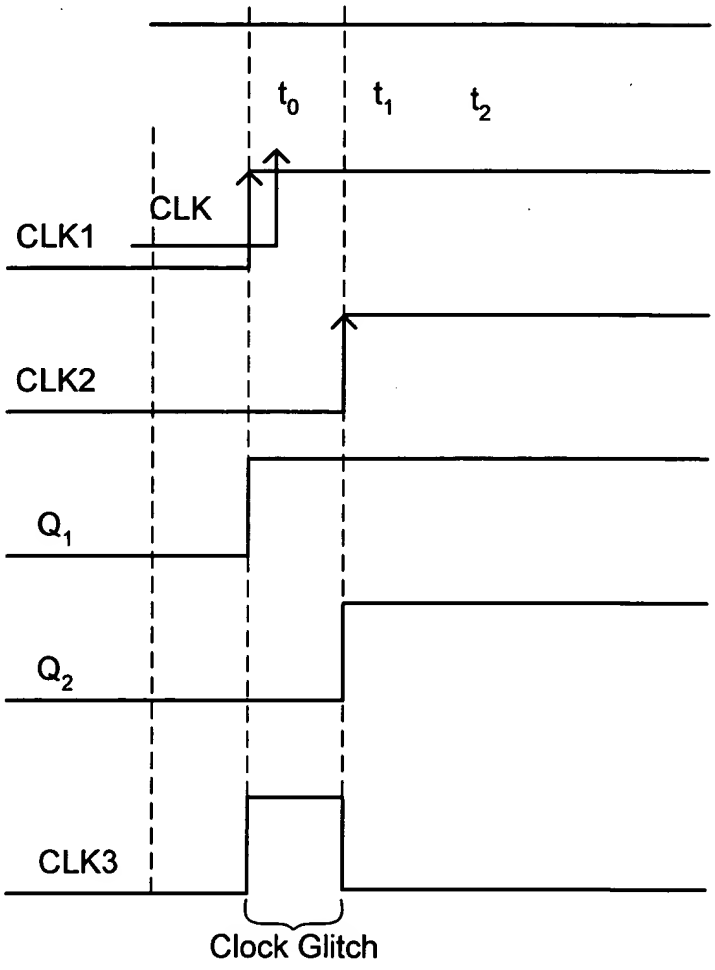


FIG. 77(B)

TIMING ADJUSTMENT BY ADDING DELAY

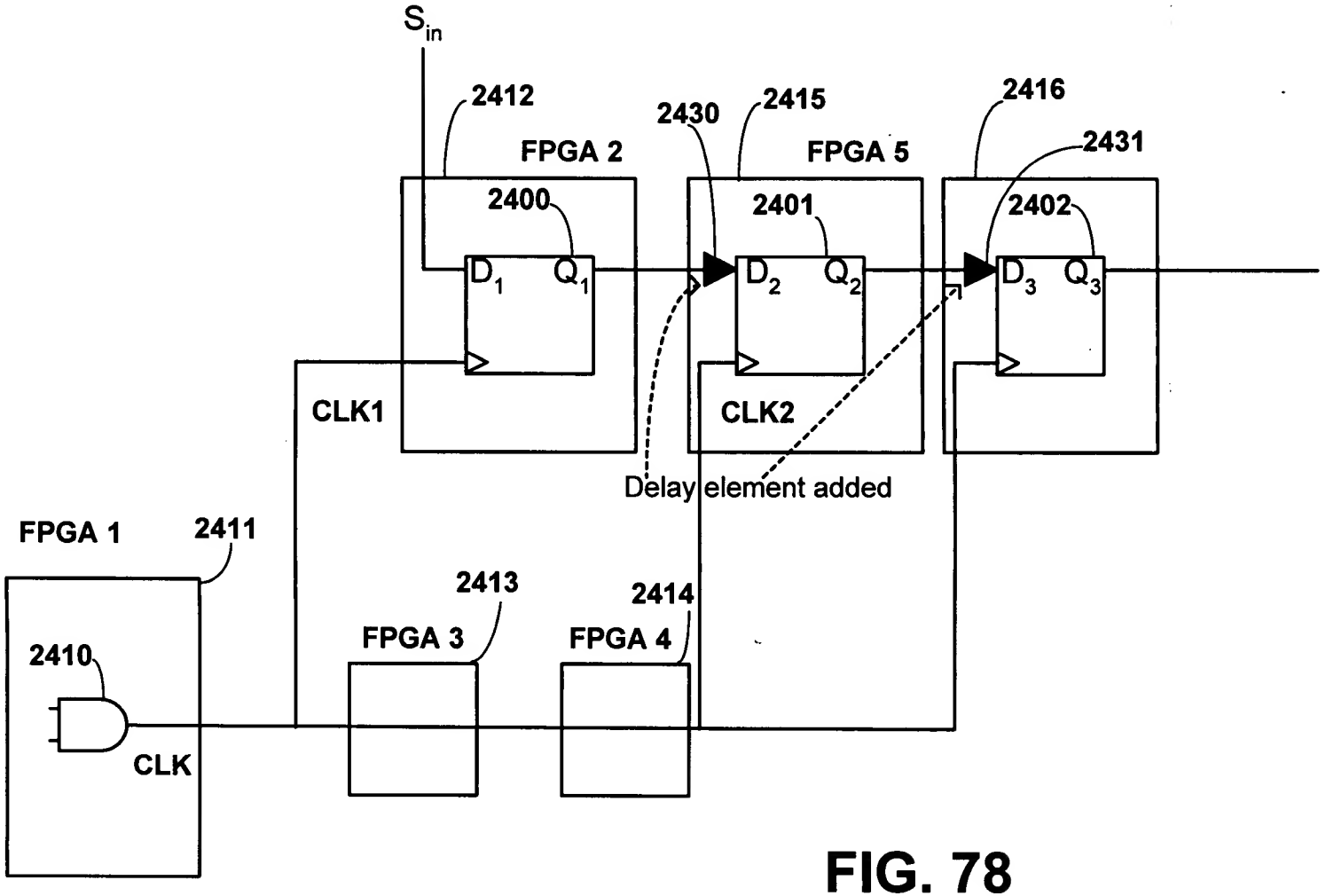
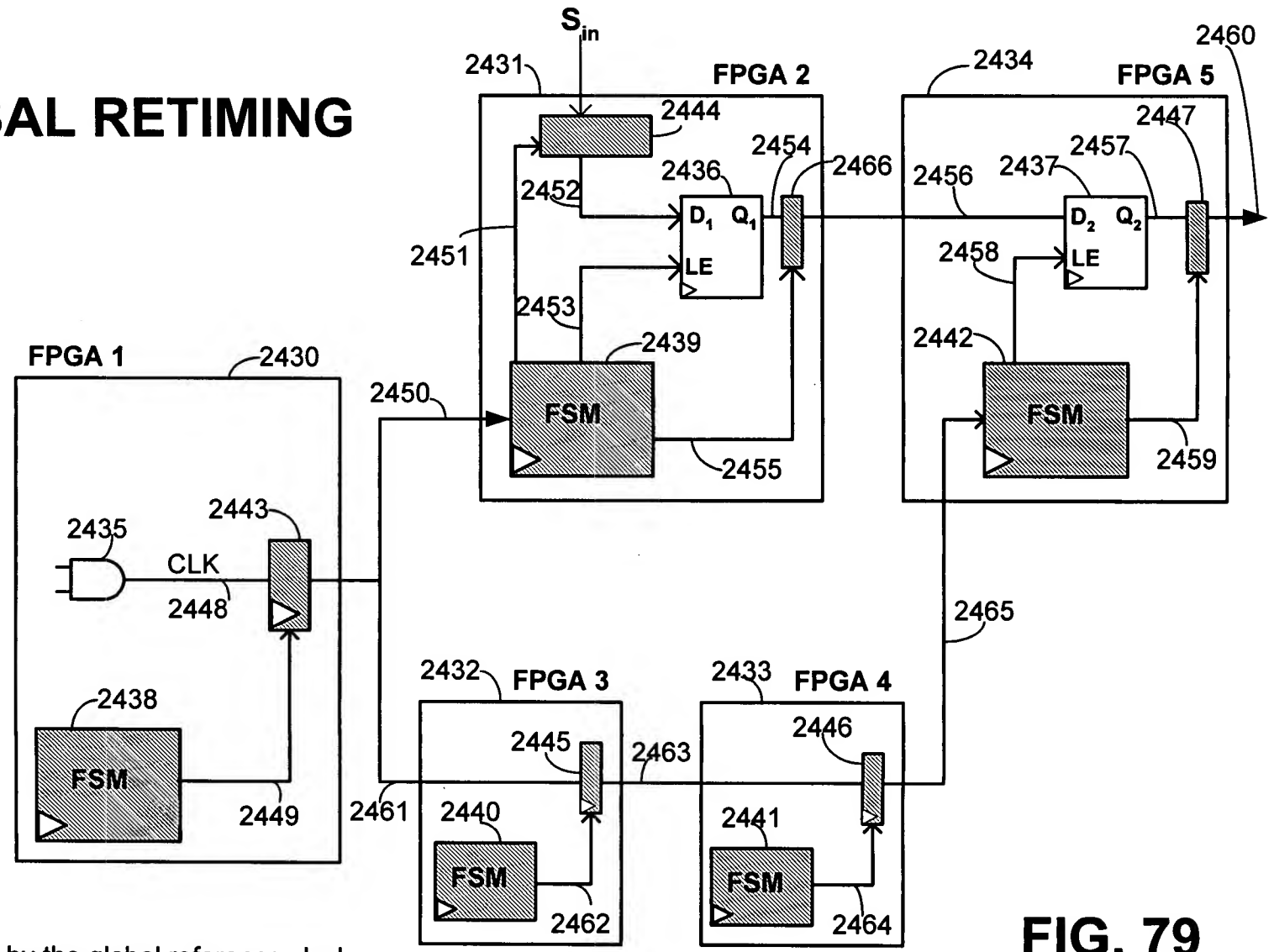


FIG. 78
(Prior Art)

GLOBAL RETIMING



Legend

Controlled by the global reference clock.

FSM and I/O registers for retiming control.

FIG. 79
(Prior Art)

TIGF LATCH

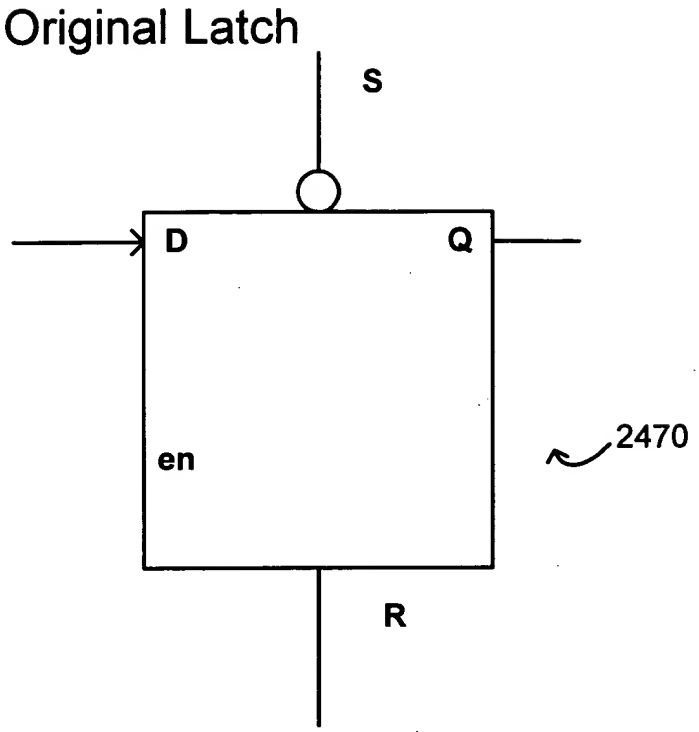


FIG. 80(A)

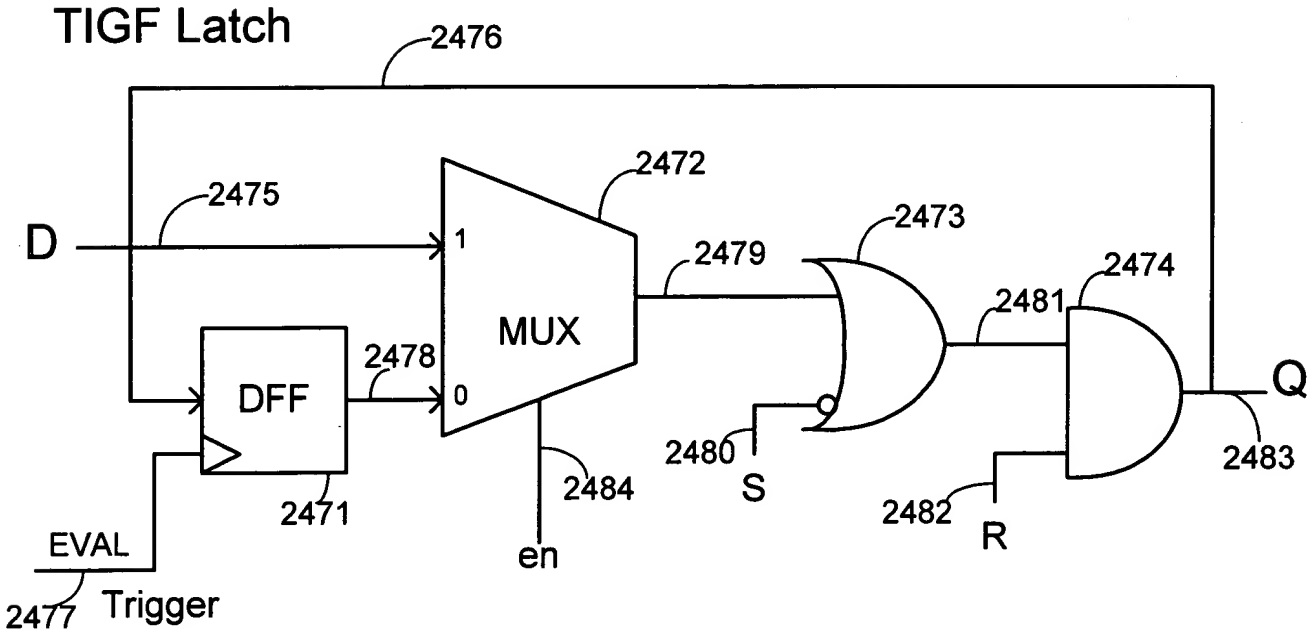


FIG. 80(B)

Original DFF

TIGF DFF

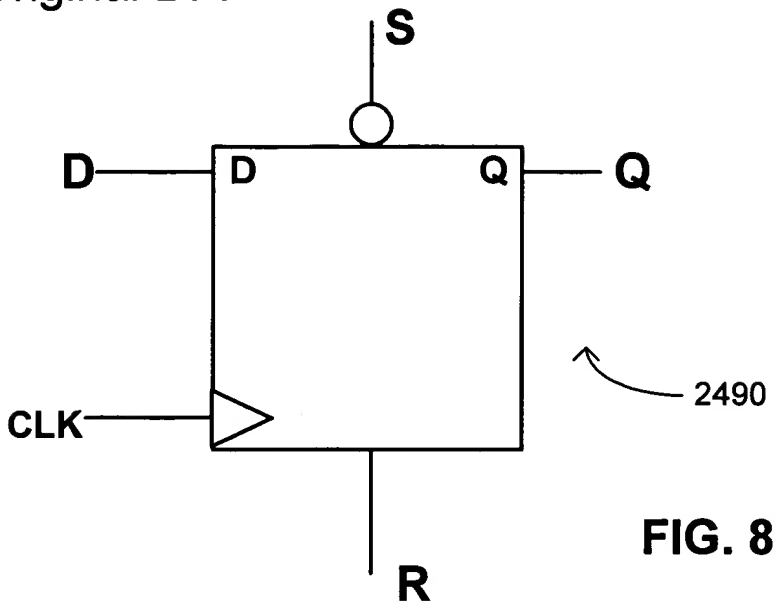


FIG. 81(A)

TIGF DFF and Edge Detector

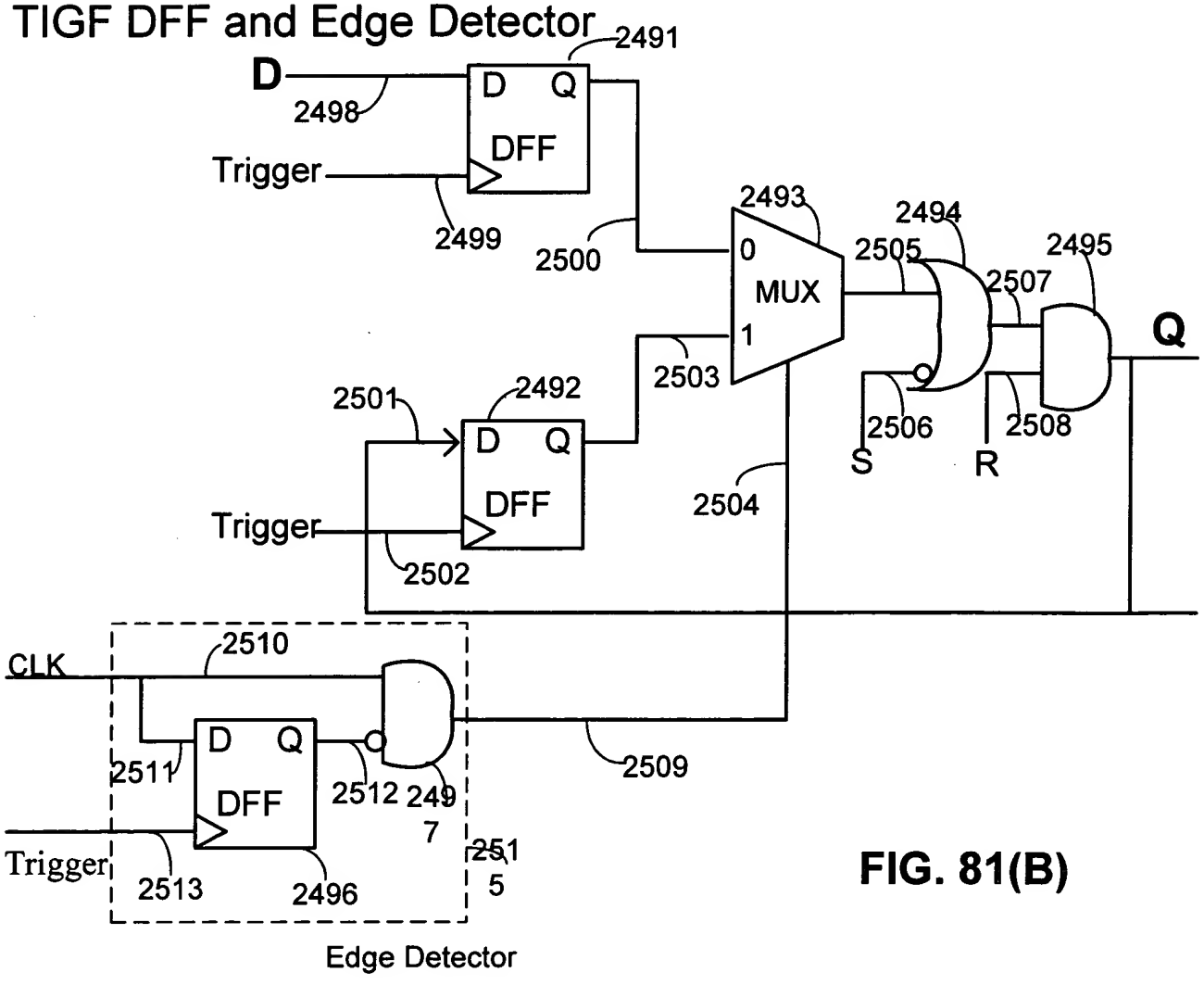


FIG. 81(B)

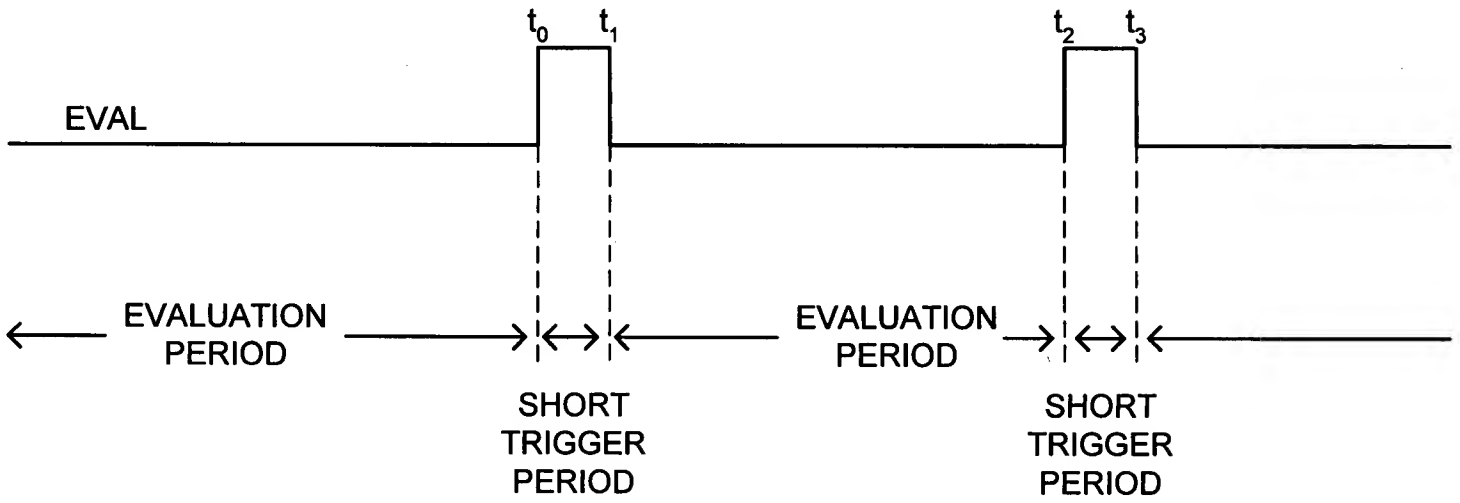

GLOBAL TRIGGER SIGNAL
t_0
t_1
t_2
t_3
EVAL
EVALUATION PERIOD
EVALUATION PERIOD
SHORT TRIGGER PERIOD
SHORT TRIGGER PERIOD

FIG. 82

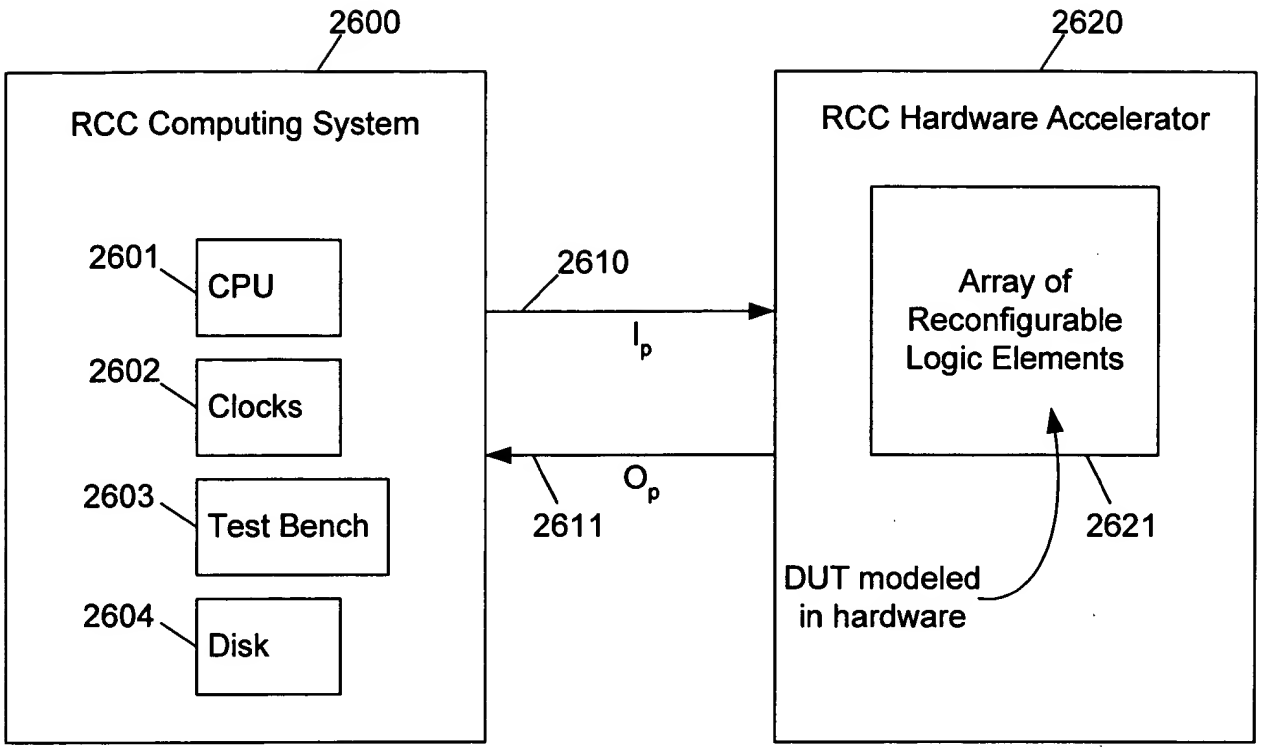


FIG. 83

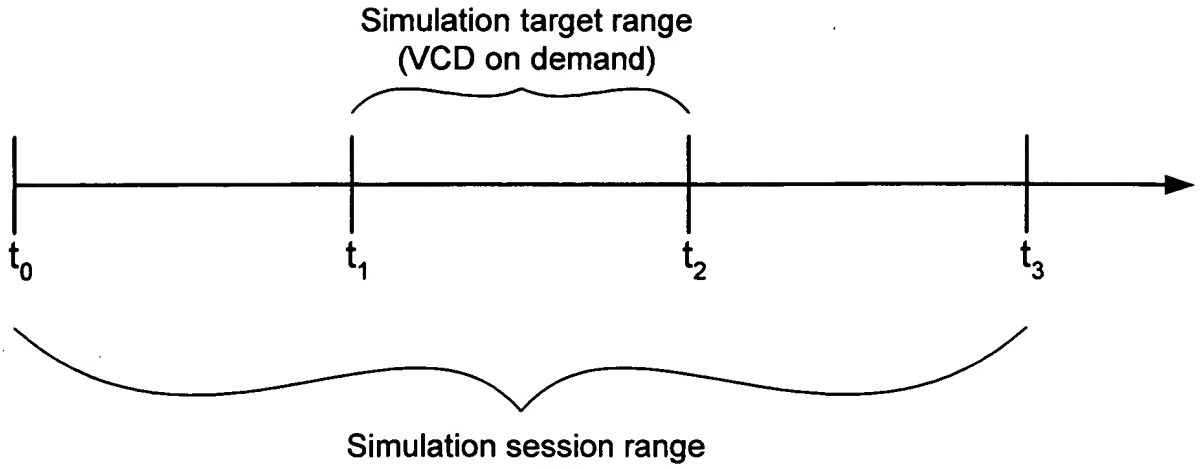


FIG. 84

SINGLE-ROW FPGA PER BOARD

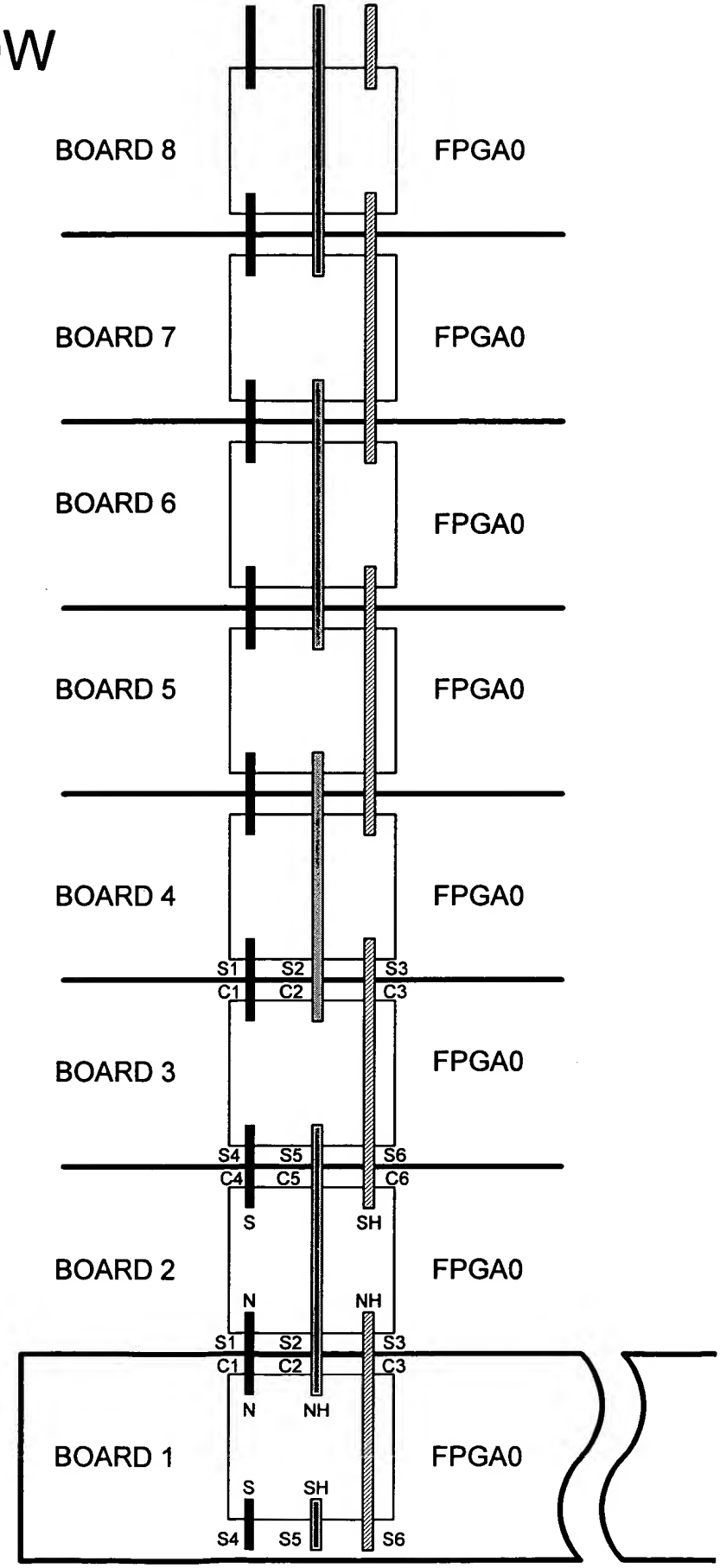


FIG. 85

TWO-ROW FPGA PER BOARD

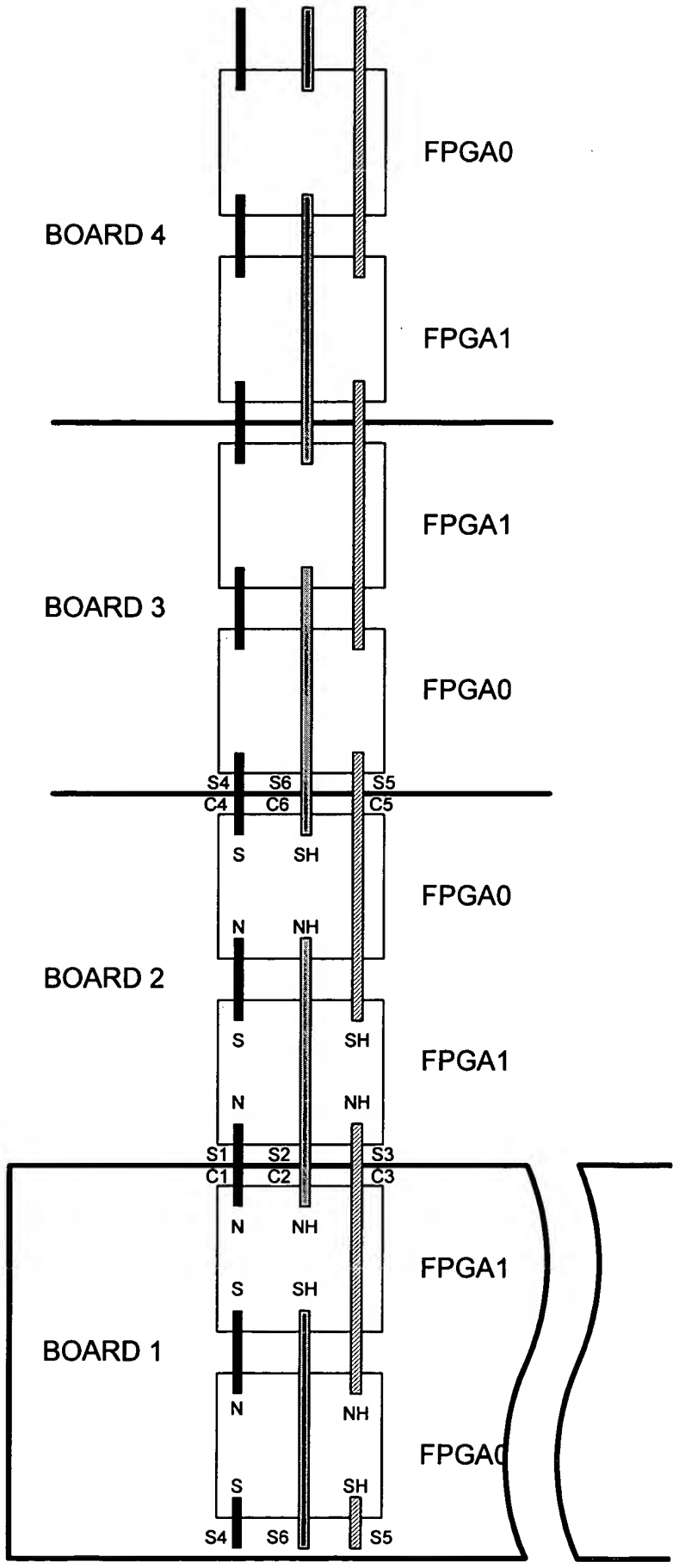


FIG. 86

THREE-ROW FPGA PER BOARD

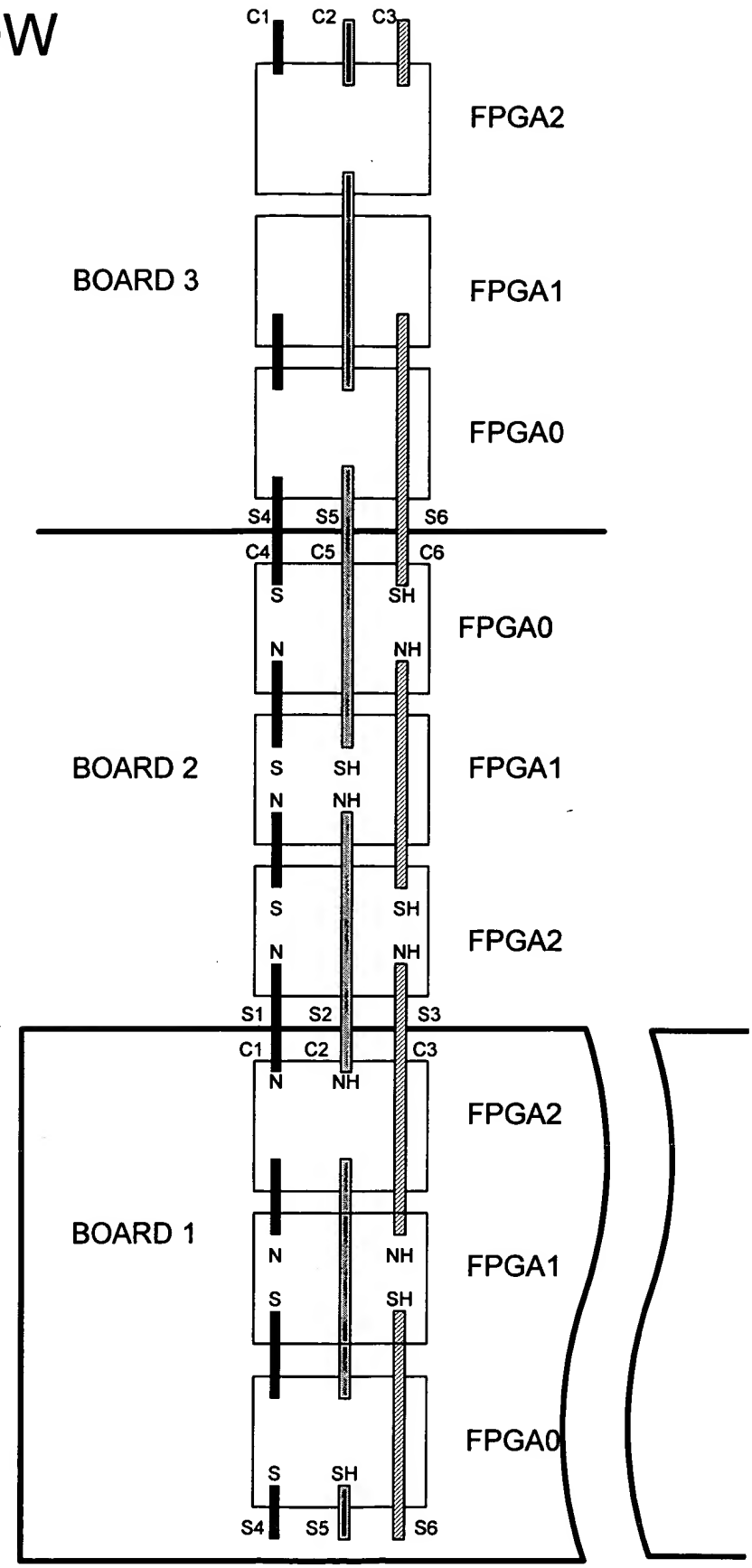


FIG. 87

FOUR-ROW FPGA PER BOARD

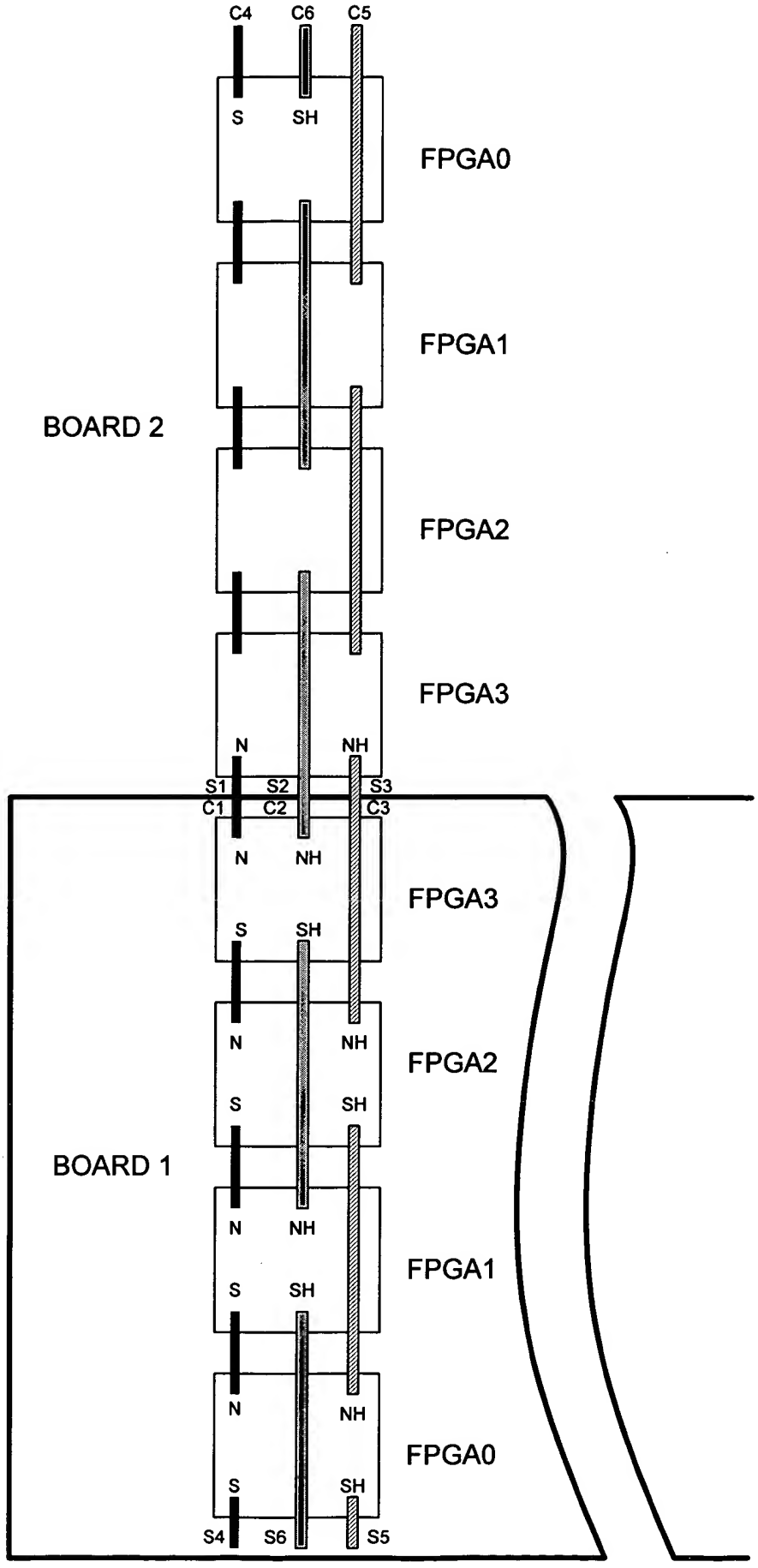


FIG. 88

INTERCONNECT FOR THREE-ROW PER BOARD

I/O Signals	Odd Board	Even Board	Common Board
	Connector-Group Pin-position	Connector-Group Pin-position	Connector-Group Pin-position
FPGA2_N	C1	S1	C1, S1
FPGA2_NH	C2	S3	C2, S3
FPGA1_NH	C3	S2	C3, S2
FPGA0_S	S4	C4	C4, S4
FPGA0_SH	S5	C6	C6, S5
FPGA1_SH	S6	C5	C5, S6

FIG. 89

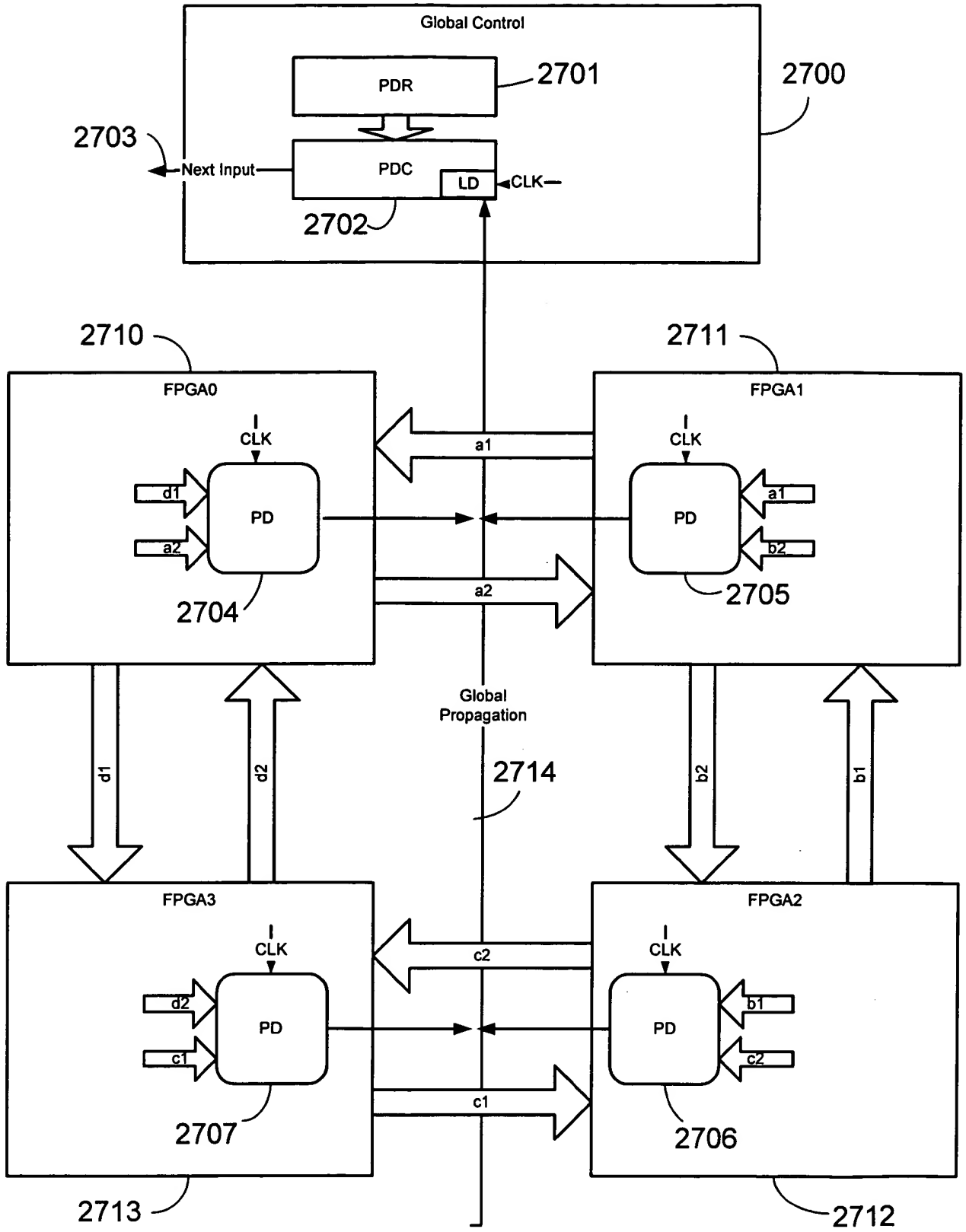

Global Control
PDR
2701
2700
2703
Next Input
PDC
LD
CLK
2702
2710
2711
FPGA0
FPGA1
CLK
d1
a2
PD
a1
b2
2704
2705
Global
Propagation
2714
d2
b2
b1
FPGA3
FPGA2
c2
d2
c1
b1
2707
2706
2713
2712

FIG. 90

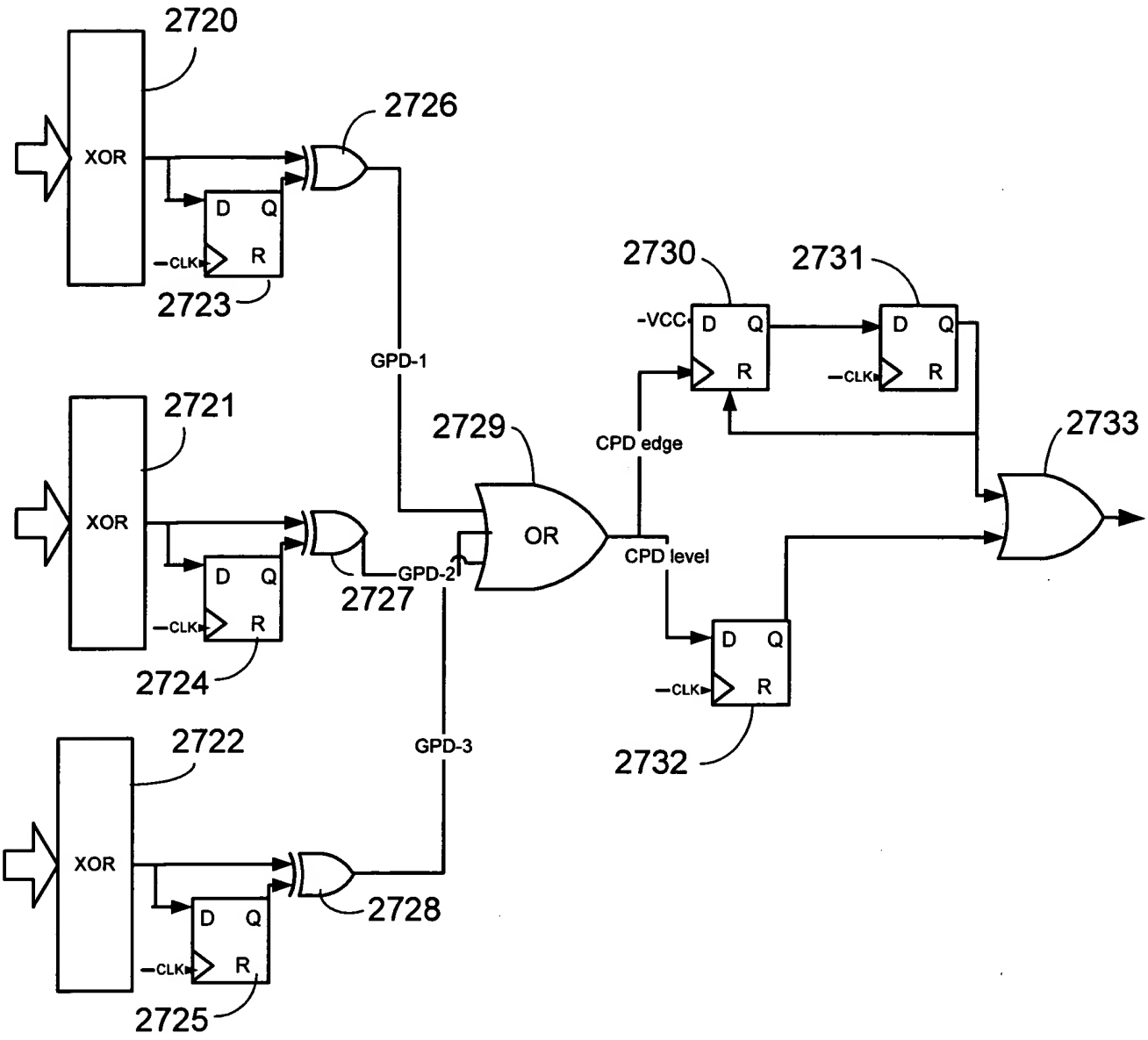

2720
XOR
2726
D Q
CLK R
2723
GPD-1
2721
XOR
2729
OR
2730
-VCC
D Q
R
2731
D Q
CLK R
2733
CPD edge
D Q
R
GPD-2
2727
CPD level
CLK
2724
2722
XOR
GPD-3
D Q
CLK R
2728
2732
2725

FIG. 91

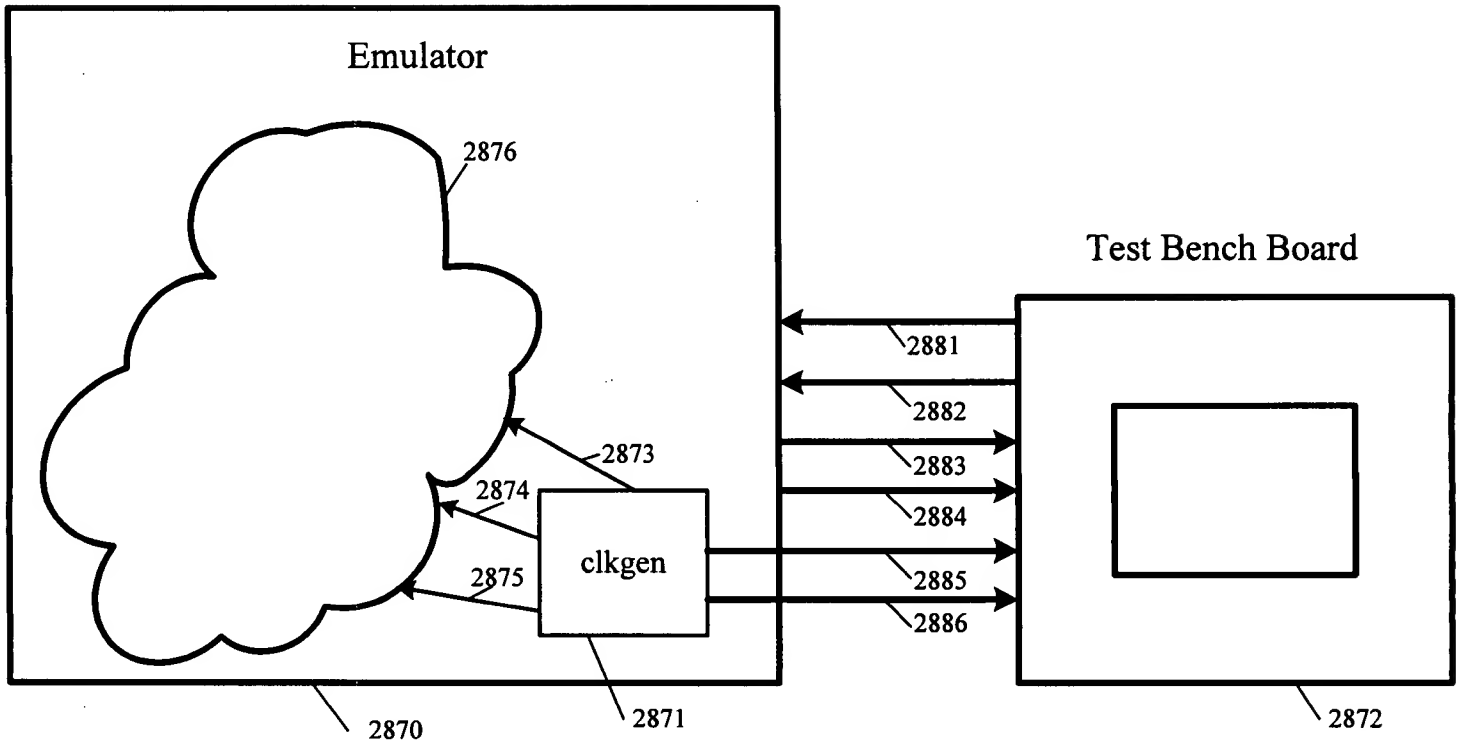

Emulator
2876
Test Bench Board
2881
2882
2883
2884
2885
2886
2873
2874
2875
clkgen
2870
2871
2872

FIG. 92

Clock Specification

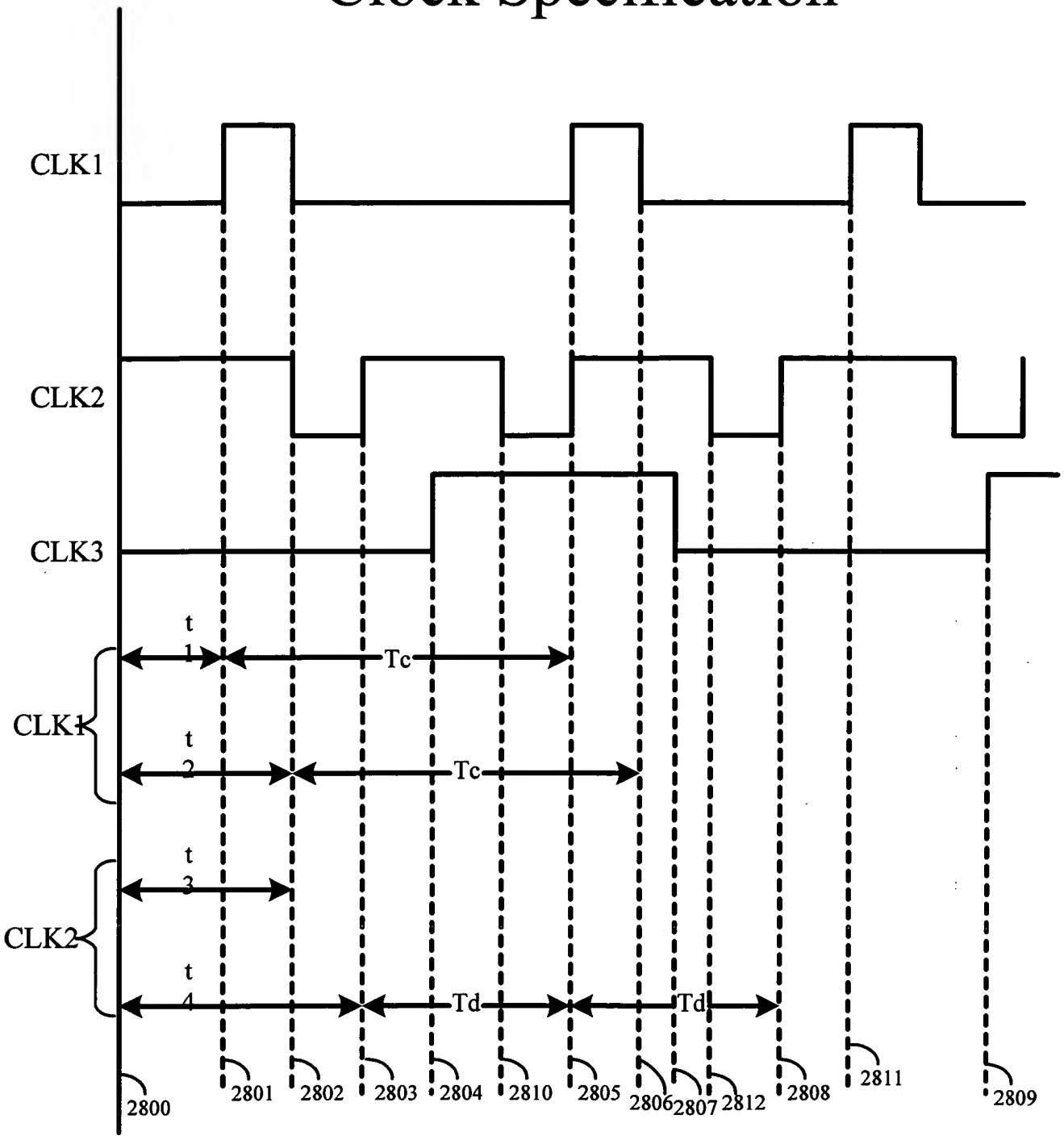


FIG 93

Clock Generation Scheduler w/ Slices

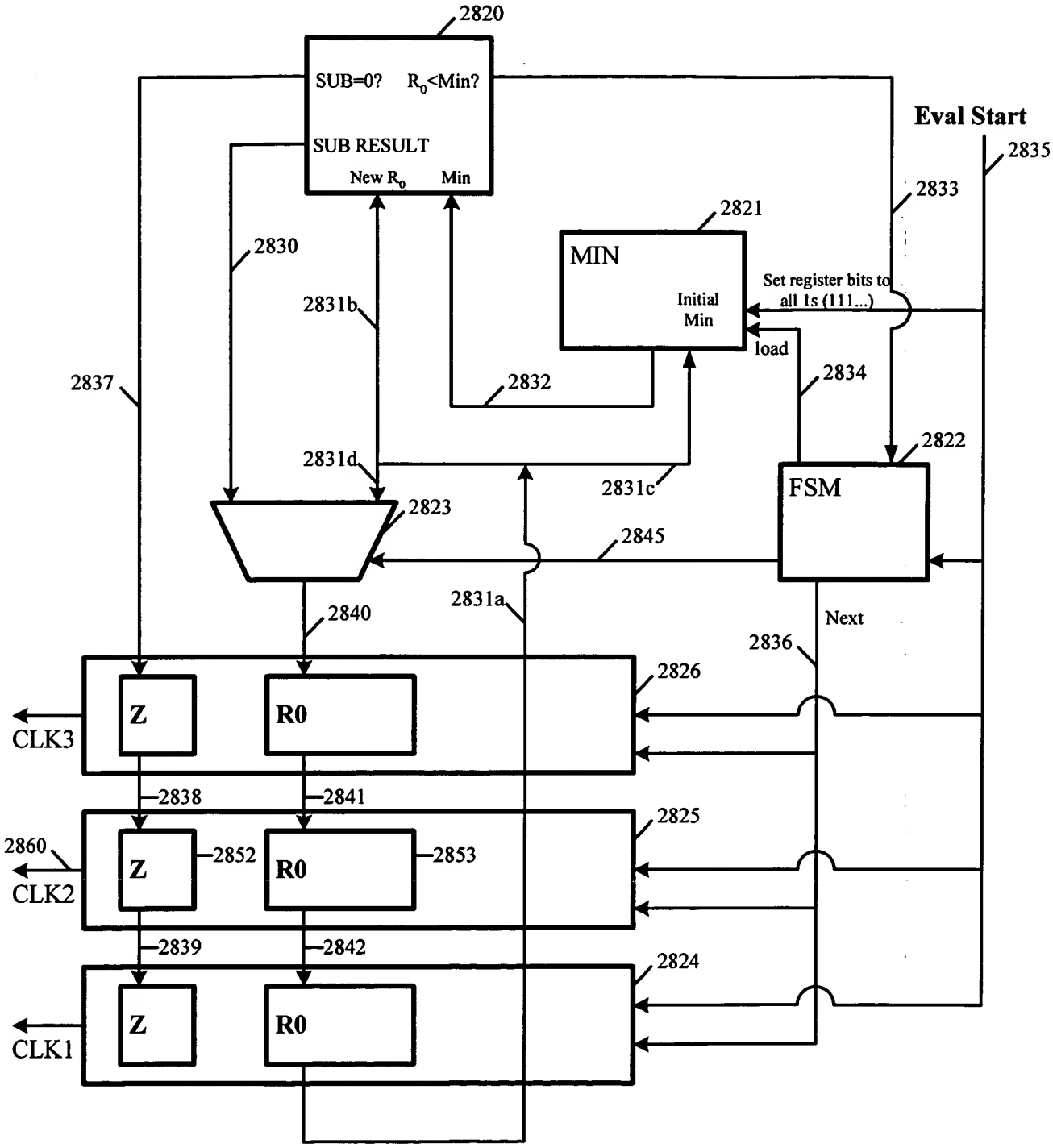


FIG 94

Clock Generation Slice

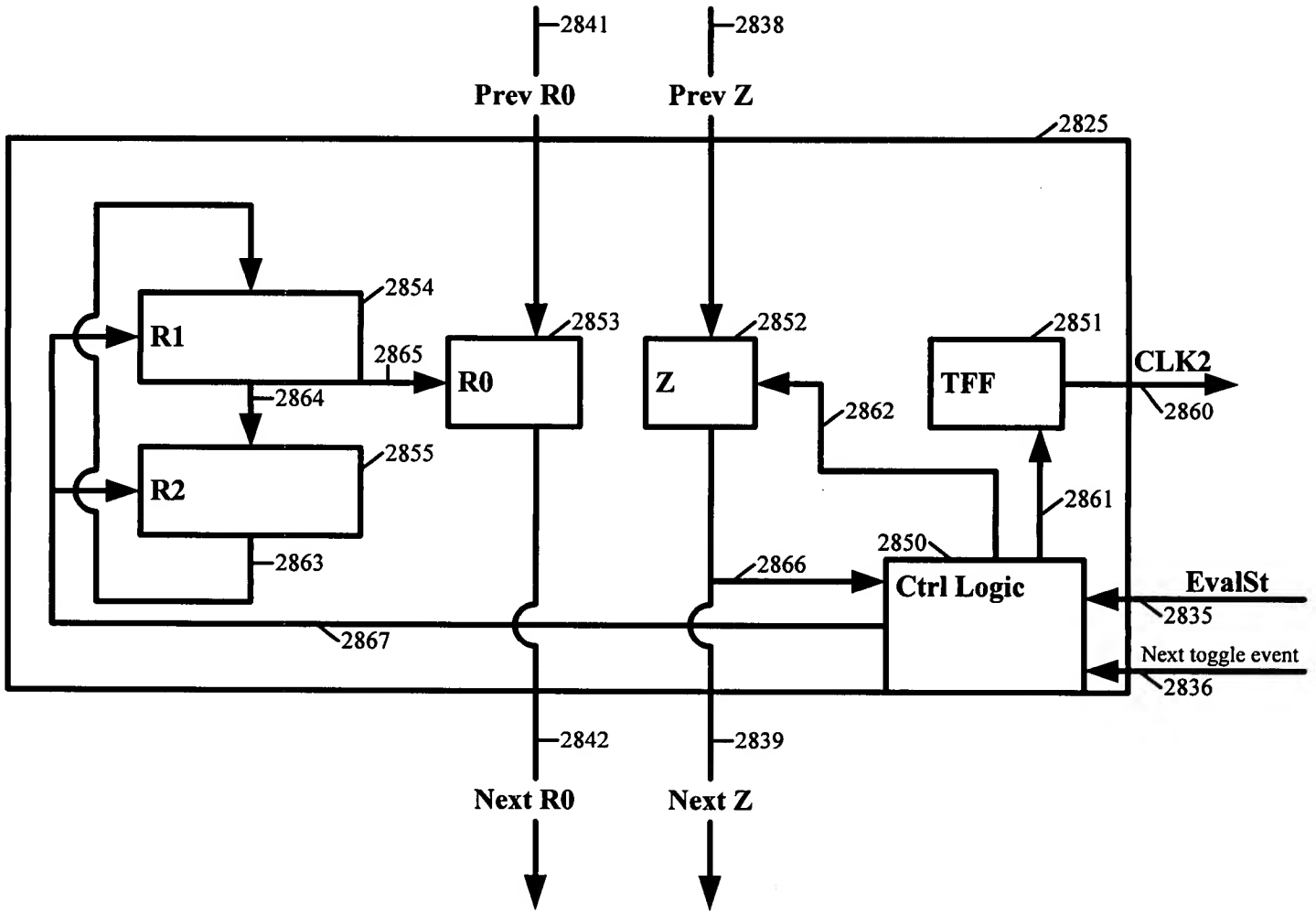


FIG 95

Clock Generation Scheduler and Slices

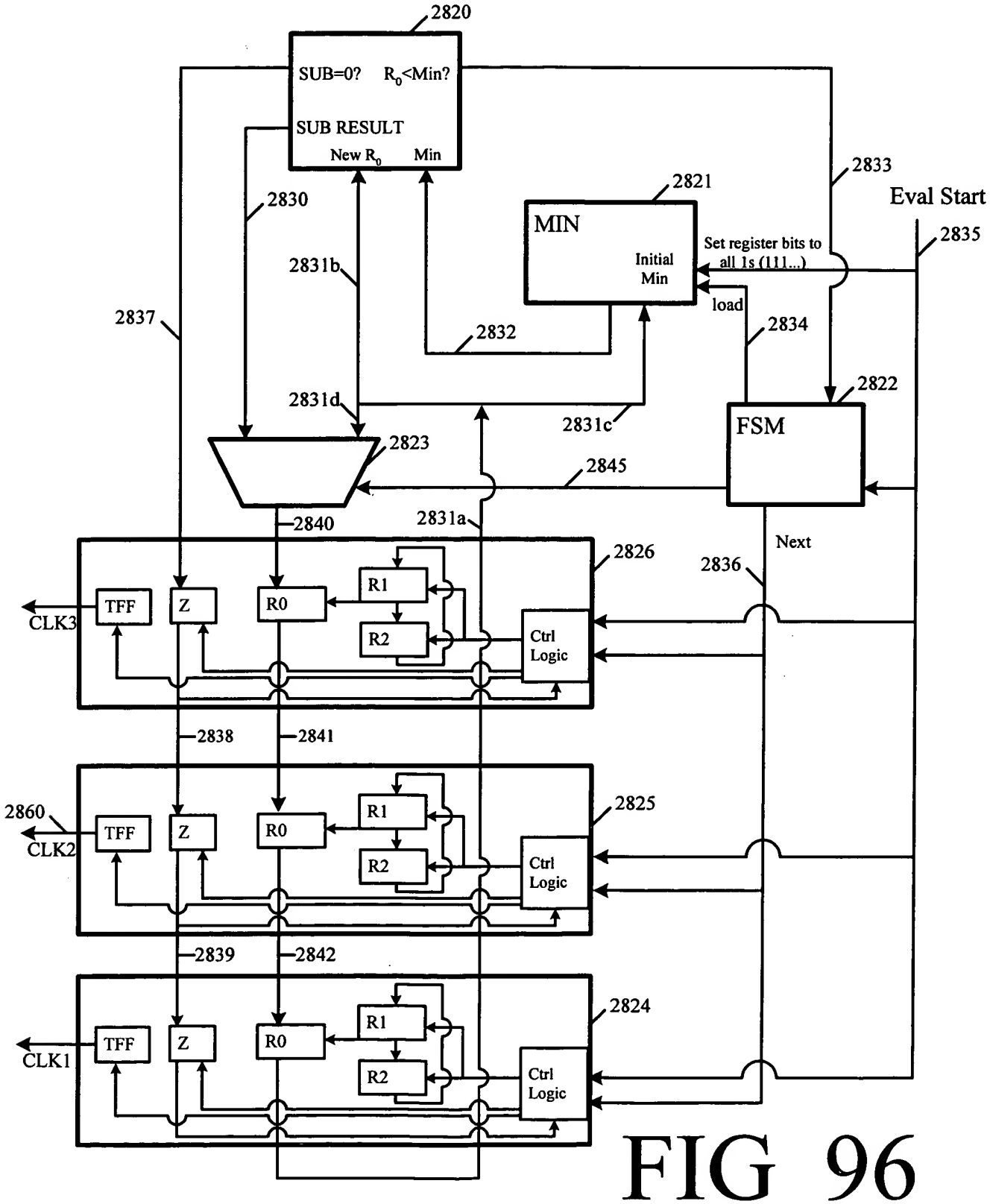


FIG 96